Management of the Internet and Complex Services

European Sixth Framework Network of Excellence FP6-2004-IST-026854-NoE

Deliverable 8.4
Documentation of Final Model and Evaluation of Economic Management Principles

The EMANICS Consortium

Caisse des Dépôts et Consignations, CDC, France
Institut National de Recherche en Informatique et Automatique, INRIA, France
University of Twente, UT, The Netherlands
Imperial College, IC, UK
Jacobs University Bremen, JUB, Germany
KTH Royal Institute of Technology, KTH, Sweden
Oslo University College, HIO, Norway
Universidat Politecnica de Catalunya, UPC, Spain
University of Federal Armed Forces Munich, CETIM/UniBwM, Germany
Poznan Supercomputing and Networking Center, PSNC, Poland
University of Zürich, UniZH, Switzerland
Ludwig-Maximilian University Munich, LMU, Germany
University College London, UK
University of Pitesti, UPI, Romania

For more information on this document or the EMANICS project, please contact:

Dr. Olivier Festor
Technopole de Nancy-Brabois — Campus scientifique
615, rue de Jardin Botanique — B.P. 101
F—54600 Villers Les Nancy Cedex
France

Phone: +33 383 59 30 66
Fax: +33 383 41 30 79
E-mail: <olivier.festor@loria.fr>

Document Control

Title: Documentation of Final Model and Evaluation of Economic Management Principles

Type: Public

Editors: Martin Waldburger, Burkhard Stiller

E-mail: waldburger@ifi.uzh.ch, stiller@ifi.uzh.ch

Authors: Mark Burgess, Marinos Charalambides, Frank Eyermann, Hasan, Javier Rubio Loyola, Samah Bel Haj Saad, Thomas Schaaf, Joan Serrat, Burkhard Stiller, Martin Waldburger (alphabetic order)

Doc ID: D8.4-v1.0

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	November 4, 2008	Burkhard Stiller (BS), Martin Waldburger (MW)	Document structure, ToC, Distribution of responsibilities
V0.2	December 4, 2008	BS, MW, Thomas Schaaf (TS), Mark Burgess (MB), Joan Serrat (JS), Frank Eyermann (FE), Marinos Charalambides (MC), Stylianos Georgoulas (SG)	ASAM, SAPDoGS, BP3EM, PRIPOL, GridAcc, Overall Economic Management Model
V0.3	December 4, 2008	BS	Inclusion of TODO and completion hints for ALL partners
V0.4	December 19, 2008	Hasan (HH), Fabio Hecht (FH), Samah Saad (SS), TS, MW, BS	Updated content in ASAM, SaPDoGS, BP3EM, GridAcc, Overall Economic Management Model, Annex, Abbreviations, References
V0.5	December 22, 2008	MW, BS	Inclusion of new updates, finalization of introductory and concluding sections as well as Annexes, corrections
V0.6	December 23, 2008	TS, SS, JS, MW, BS,	Inclusion of new updates, finalization of sections, References, corrections
V1.0	December 24, 2008	BS, MW	FInalization and issued as a Christmas Gift to WP8 as well as a hand-in to the project coordinator for submission to the commission.

Table of Content

1 Executive Summary

Modeling of economic principles in a network management context has been addressed in the EMANICS work of work package WP8 so far. Such an approach undertaken leaves the purified grounds of technical work only and does lead to a generic information infrastructure enabling multiple parties' access — even between *multiple domains* — in an economically maintained manner. Thus and in particular, the economic dimension of network management for Internet Service Providers (ISP) has been an inherently integrated part of an IP-based network solution until now, however, the main and side effects of this situation have been partially underestimated in an operational environment.

To be able to overcome this problem the EMANICS' work package WP8 on "Economic Management" decided to investigate a number of important aspects of such economically driven network management and network operations-based mechanisms. These include

- Auditing for Service Level Objectives (SLO) Across Provider Domains [ASAM],
- Service Level Agreements and Promise Descriptions for Grid Services [SaPDoGS],
- Best Practices, Processes, and Promises in Economic Management [BP3EM],
- Pricing by Policies [PRIPOL], and
- Grid Accounting [GridAcc].

For each of these areas a set of new insights, simulation models, and implementations has been achieved, which are intended for a future application in the networking domain outside of EMANICS. The key findings are summarized as follows:

Due to the fact that the allocation of distinct auditing roles of participating communication entities depends on data streams — varying from application to application —, they have to be audited securely in a commercial environment. Based on measurements within ASAM four different traffic profiles have been identified, resulting in a detailed protocol sequence description. Thus, a simulation of multi-domain auditing as well as an implementation of the SLO Compliance Monitor are undertaken and proposed.

SaPDoGS has investigated the use of promise theory and the voluntary cooperation paradigm to understand SLAs in organizations generally, with a particular look at Grid services. Now, the SLA is a concept being used loosely, mainly in contract terms between providers and customers. Promises, *e.g.*, between agents, can deal with aspects such as Quality-of-Service, quality of behavior, or specifications of state. The methodology developed for the Grid services use case allows for the performing of a rigorous accounting of uncertainties associated each with separate promises.

Best practice recommendations are paid a larger attention within IT management. As well, this holds true for process-oriented approaches, like the IT Infrastructure Library (ITIL). This turn from a purely technological point of view and from a perspective covering organizational aspects into a more complete IT Service Management (ITSM) attention comes with various challenges. Thus BP3EM contributes in bridging the gap between organizational ITSM approaches as well as the "hands-on-the-keyboard" system management approaches and administration tasks.

While policy-based networking defines an approach to control the quality level a network is able to deliver, PRIPOL offers the additional application of policies to be used for determining the pricing of services offered. Typically, those policies are stored in directory-enabled network components, which show a quite large complexity, however, they are not fully standardized yet. This openness offers the opportunity PRIPOL is taking: PRIPOL

offers an approach to modify these policies at run-time to achieve an ongoing modeling of business values, driven by current business demands of, *e.g.*, operators.

Large-scale service-oriented computing in a local computing center context of a computational Grid or across multiple domains within dynamic VOs (Virtual Organization) clearly states the need for the suitable economic management mechanisms to be in place. Accounting of services provided and resources used constitutes a management mechanism of key importance. While neither existing approach showed a sufficient support of multi-domain or virtualization concepts nor any approach was based on suitable accounting principles from business domains, these gaps identified led to the development of an accounting model suitable for computing centers and dynamic VOs. This was applied in the context of the Leibniz Supercomputing Centre and derived viable results and optimization information.

A special note on the clear success of EMANICS' WP8 has to be named explicitly: A Journal Paper appeared on the GridAcc approach of WP8, explicitly on "*Evaluation of an Accounting Model for Dynamic Virtual Organizations*", Journal on Grid Computing, Springer. Additionally, the new approaches of applying promise theory in economic management from the SaPDoGS approach has resulted in a key note speech on "*The Promise of Self-Adapting Equilibrium*" at the 5th IEEE International Conference on Autonomic Computing (ICAC 2008). Finally, with respect to the discussion of technical mechanisms in support of incentives, the IETF has started — at least under the consideration of Peer-to-peer (P2P) traffic — to establish a working group (ALTO: Application-layer Traffic Optimization), where one EMANICS WP8 partner is involved in, in collaboration with another FP7 project (SmoothIT).

2 Introduction

Modeling of economic principles in a network management context and managing the Internet as well as its related services and traffic determines the important goal for any operational network. This has become even more obvious due to commercial interests in service provisioning. Thus, EMANICS in general and its Work Package 8 (WP8) on Economic Management specifically intends to understand, to determine interrelations, and to simplify such Internet-based network management tasks.

In continuation of the previous project year, this area of work has been addressed in the at hand Deliverable D8.4 explicitly under an integrated point of view (cf. Figure 1) of three major methodological points of view, determining the scope:

1 the multi-domain aspect,

2 the optimization of existing IT service management approaches, and

3 concrete economic or economics-supporting mechanisms.

As outlined in Table 1 topic 1 and 3 are combined to the approach on multi-domain auditing (ASAM), the Service Level Agreement (SLA) application on Grid services within the context of promise theory addresses topic 2 (SaPDoGS). The BP3EM approach outlines key gaps between IT service management and system administration, which addresses topic 2. PRIPOL offers an alternative path to determine the market price of services based on policies, addressing topic 3. The combination of topic 2 and 3 is undertaken by GridAcc approach, which combines the concrete accounting mechanism for Grids as an optimization for computing centers and their cost structure.

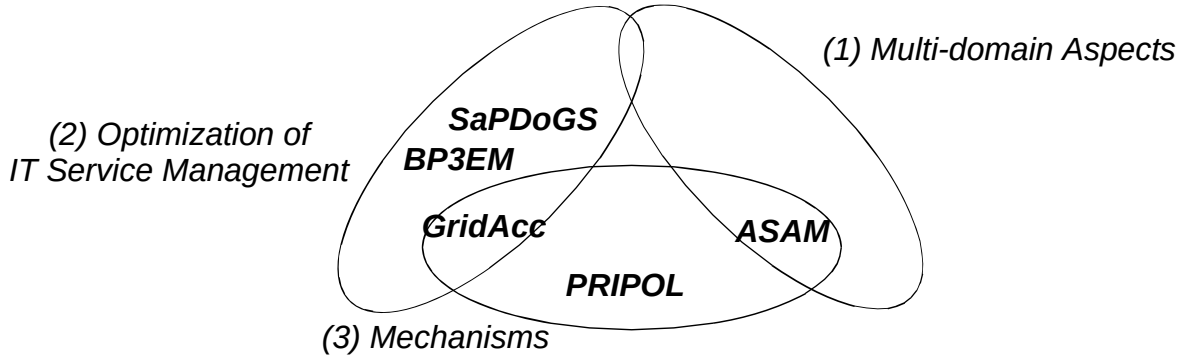


Figure 1: Remaining Scope of D8.4 Dealt by within 5 EMANICS WP8 Approaches [31]

Table 1: Work Package 8 (WP8) Projects, Their Durations, and Partners Involved

Acronym	Full Project Title	Start Time	End Time	Partners
ASAM	Auditing of SLOs Across Multiple Provider Domains	M+19	M+36	UniBwM, UniZH, UniS
SaPDoGS	SLA and Promise Descriptions of GRID Services	M+19	M+36	HIO, UniBwM
BP3EM	Best Practices, Processes and Promises in Economic Management	M+19	M+36	LMU, HIO
PRIPOL	Pricing by Policies	M+19	M+36	UPC, KTH, UniZH
GridAcc	Grid Accounting	M+19	M+30	UniZH, UniBwM, LMU/LRZ

Managing the infrastructure of tomorrow's Internet requires suitable technology and mechanisms as well as valid and viable economic means, which will lead to a generic information infrastructure and which enables multiple parties' access — even between

multiple domains — in an economically fair manner. In particular the economic dimension of network management for Internet Service Providers (ISP) has to be an integrated part of an overall IP-based network solution.

Thus, the previously developed EMANICS business modeling approach (cf. Figure 2) is consulted here in order to determine the respectively applicable role, service, and charging model characteristics for each of these presented 5 WP8 projects. This facilitates not only a structured project assessment in terms of common business modeling dimensions, it also lays down the basis for an integrated viewpoint on how and where each project contributes to each other, and finally to a well maintained overall economic management model going beyond current state-of-the-art.

While both aspects are fully documented in Section 8 and partly in Section 9, the first aspect of an assessment according to the EMANICS business modeling dimensions focuses in the area of role models on which actors a WP8 project addresses and in which market segments these operate. This may include here ISP markets and markets of service providers as well as operators of large computing centers. With regard to service model dimensions, each project is considering what commercially exploitable value-added it contributes to those addressed actors and market segments and in which cooperation structure value is created.

Finally and driven by the overall target of an economically sustainable management for tomorrow's Internet, the apparent remaining gap to a sustaining business is bridged by an analysis of each project's opportunities in profit, in its cost structure, and with regard to the adopted competitive strategy.

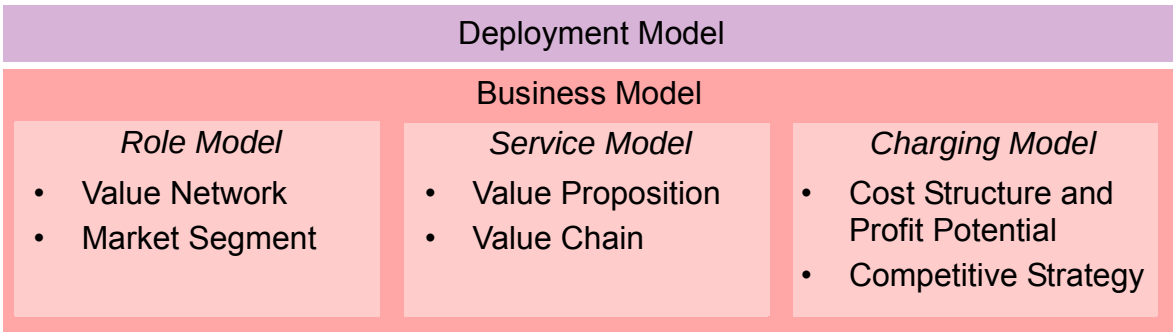


Figure 2: EMANICS Business Modeling Overview [33]

2.1 Document Outline

This deliverable D8.4 is organized as follows, which is fully in-line with the Phase III project-based operation of EMANICS. Section 3 describes the ASAM approach and its findings in measurements and implementation. Section 4 addresses SaPDoGS' application of promises onto Grid services. Section 5 outlines all major findings and methods applicable in the domain of IT Service Management. Section 6 discusses the current state of the approach, which applies policies to determine prices of services offered. Section 7 describes the accounting model developed for Grid services and applies it to the LRZ use case of a real-life computing center.

Section 8 summarizes those insights obtained under the common view of economic management principles and outlines the potentials of each of these pieces of knowledge gained, if it is combined in a practical environment at a later stage. Finally, Section 9 summarizes D8.4, draws conclusions, and gives an outlook to next steps. Furthermore,

Section 10 provides the list of explanations of major terms applied. Section 11 depicts the key bibliographic references, which have been considered. Section 12 lists all abbreviations utilized within D8.4.

Additionally, Section 14 summarizes the output of a promise theory workshop organized by one the WP8 partners. Section 15 summarizes the business-driven IT management workshop. Section 17 provides a report on the 73rd IETF meeting, with the focus on economic management principles applied onto overlay traffic. Last but not least, Section 18 lists and summarizes all research papers done in the context of EMANICS WP8.

3 ASAM: Auditing of SLOs Across Multiple Provider Domains

In [33] a multi-domain auditing system was designed. The initial system design shown there is refined in this document. First a basic scenario is described and roles in an auditing scenario are identified.

The allocation of roles depends on the data streams, which should be audited. These data streams vary from application to application. Section 3.2 analyzes different applications and their signature data streams. Four different so called traffic profiles are identified. In Section 3.3 the detailed protocol sequence for each identified traffic profile is described. Because the signature data streams are different, role allocations vary. While Section 3.4 presents a simulation of multi-domain auditing, Section 3.5 proposes an implementation of the SLO Compliance Monitor. This chapter is completed with an evaluation in Section 3.6 and a summary of a complementary work in Section 3.7.

3.1 Basic Scenario and Roles

In Figure 3 an abstract scenario based on the one shown in [33] is depicted. The abstract scenario is suitable for the following discussions as the auditing system must work for a large number of scenarios. Furthermore, this section will define relevant terms for the following sections.

The designed auditing system monitors network-related parameters, i.e., performance parameters of IP packets. A sender and a receiver as source and destination of these packets, respectively, are defined. The sender is the source of a stream of packets, which he addresses towards the receiver. The receiver in turn receives and processes the packets and is therefore the sink of the packets.

Sender and receiver are connected each to the network of an Internet Service Provider (ISP), who operates a network of links as well as links to other ISPs. The detailed topology within the network of an ISP, i.e., which router exists and how they are connected, is irrelevant for the auditing process, because the auditing system is only required to determine the domain of an SLA violation. A finer localization of the violation is not necessary. Therefore, only domain boundaries and connections between domains are shown in this scenario.

The role of the sender and receiver are valid for a single unidirectional flow only. For the notion of “flow” several definitions exists. A broad definition can be found in [29], which groups packets according to a set of common header fields. While this definition is valid from the perspective of the IP Flow Information Exchange (IPFIX) working group, for an auditing system it is too coarse. This definition allows aggregation, which is not acceptable

for an auditing system, because an application should be able to control the auditing system and the paths to different receivers could show different performance values.

Taking these considerations into account, a flow should be defined as an application-level end-to-end stream, where all packets exhibit a common traffic profile. However, it is hard to work with this definition in practice, because detailed knowledge of each application would be necessary in order to identify the flows and assign packets to a flow. The definition could be reformulated under the assumption that each application uses different TCP or UDP ports for different application-level end-to-end stream; an assumption which is typically valid for most applications. Thus, a flow is defined as a series of packets which share the same quintuple of source address, destination address, source port, destination port, and protocol.

Figure 3 shows a topology based on this model. The sender is connected to the network of ISP A, which has links to ISP B and ISP C. The receiver is connected to ISP D, which has links to ISP C and ISP E. Three other ISPs (ISP B, ISP C, and ISP E) are shown in this scenario. They have connections to the other ISPs and traffic might be routed over their networks, if the connection between ISP A and ISP C, or between ISP C and ISP D is not operational. Border routers are implicitly assumed at the end of each inter-domain link.

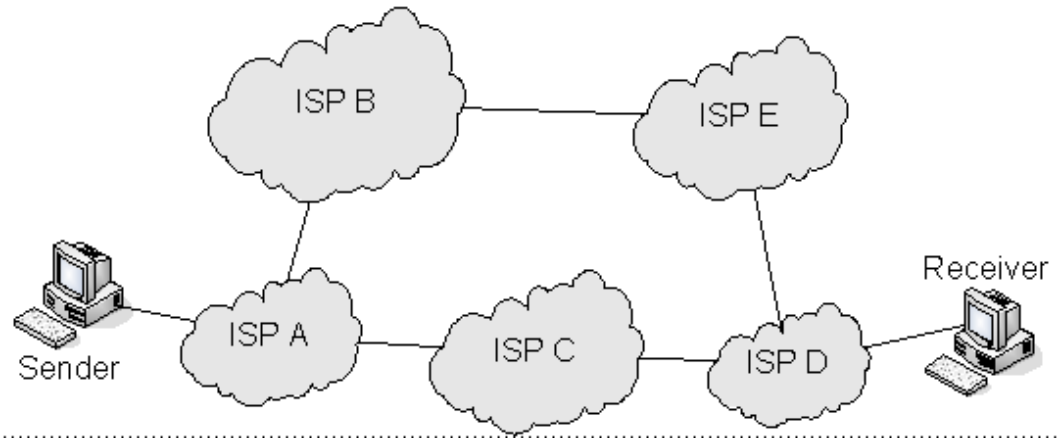


Figure 3: Simplified Internet Scenario

Looking from the perspective of the auditing system three more roles could be identified. First of all the customer is the one who requests auditing for a connection. Either the sender or receiver could be the customer as shown in the next sections. The customer needs to have an SLA with his Access Network Provider. It is the contractual partner who guarantees compliance with the agreed upon SLOs. The performance guarantees are the basis for performance measurements and violation detection.

All other ISPs which take part in the service provisioning are called Transit ISPs and form the core network. They do not have any direct contractual relationship with the customer, nor do they know any details of the relationship between the customer and its Access Network Provider.

3.2 Traffic Profiles

An auditing system needs to support different types of applications with their respective traffic profiles. Those different types of traffic profiles are categorized in this section according to the signature of their traffic flows.

3.2.1 Peer-to-peer Traffic Profile

A peer-to-peer traffic profile shows two flows with opposite directions and approximately same size. Thus no upwards or downwards could be identified. Voice-over-IP is a typical example of a service with this kind of traffic profile. Please note, peer-to-peer services (as *e.g.*, file sharing services) do not need to show a peer-to-peer traffic profile as defined above.

The SLA in place includes specifications for one-way performance parameters (i.e. one-way delay and loss) in both directions, as services with peer-to-peer traffic profile require fulfillment of the performance parameters for each direction. Fulfilling only aggregated round-trip parameters is not sufficient.

Taking the Voice-over-IP example, the user initiating the communication is the caller; the one receiving the call is the callee. The assignment of the auditing system roles is shown in Table 2. As discussed above, both directions need to be audited separately; therefore caller and callee are both sender and receiver at the same time. Depending on whom - caller or callee - has an SLA with his provider, the caller or the callee is in the role of the customer. Both cases are possible: *E.g.*, a company may have an SLA which guarantees performance parameters for incoming and outgoing calls.

Table 2: Roles in a Caller-initiated Peer-to-peer Traffic Profile Scenario

	Caller	Callee
Customer	X	
Sender	X	X
Receiver	X	X

Table 2 and Table 3 show the role occupations for these both cases, respectively. The first case, where the caller has an SLA with his ISP, is termed caller-initiated peer-to-peer traffic profile, the other case callee-initiated peer-to-peer traffic profile.

Table 3: Roles in a Callee-initiated Peer-to-peer Traffic Profile Scenario

	Caller	Callee
Customer		X
Sender	X	X
Receiver	X	X

3.2.2 Client/Server Interactive Traffic Profile

The Client-Server-interactive traffic profile comprises all classical client-server applications except batch processing and bulk data transfer (see below). Typical for this scenario is that client and server frequently exchanges information, thus some kind of dialog between both entities happens. Each exchanged piece of information is assumed to be relatively small. For the client the time until he receives the answer from the server is of interest, because the client needs to interrupt its work and wait for the answer of the server to arrive. Examples include web browsing, the use of web services, or database or server requests in two- or three tier applications.

The most prominent performance parameter in this kind of scenario is the response time of the server, i.e. the time the client has to wait for the server's response. This time includes the network delay upstream (from the client to the server), the time it takes the server to process the request and generate the response, and the network delay downstream (from the server to the client).

Only the sum of the up- and downstream network delay is of interest for an auditing of network performance parameters. Measuring the time the server needs to calculate the response is not a task of an auditing system for IP carrying services. Measuring the time the server needs to calculate the response is not task of an auditing system for IP carrying services. Separate measurement of both times is not necessary, as it is irrelevant for the client how the latency in detail arises. However, the requirement that the auditing system needs to localize violations still remains. The Client-Server-interactive scenario is the only one implementing round-trip auditing.

Table 4: Roles in a Client-initiated Client-server-interactive Traffic Profile Scenario

	Client	Server
Customer	X	
Sender	X	
Receiver	X	

In scenarios with this kind of traffic profile the communication originates at the client and terminates at the client again. He occupies therefore the sender and receiver role. Customer, however, could be either the client or the server. Analog to the previous traffic profile, each case is termed client-initiated or server-initiated client-server-interactive traffic profile, respectively. Table 4 and Table 5 summarize the role occupation in both cases.

Table 5: Roles in a Server-initiated Client-server-interactive Traffic Profile Scenario

	Client	Server
Customer		X
Sender	X	
Receiver	X	

3.2.3 Server Streaming Traffic Profile

Another important traffic profile for auditing is Server-streaming. Quite similar to the one above it is a client-server scenario, however, with extremely asymmetric packet streams. It is characterized by no or almost no upstream traffic, but a continuous packet stream in the downstream direction. The information sent upstream by the client is only used to control the packet stream sent by the server downstream.

In contrast to the previous scenario no “dialog” happens between client and server. The client once requests data, which starts the sending process on the server, and the server keeps sending data until the client requests a termination or the data (requested by the client) was sent completely. Server streaming is a potential multicast scenario. The data sent from the server can be replicated by multicast-enabled routers in the Internet.

Table 6: Roles in a Client-initiated Server Streaming Traffic Profile Scenario

	Client	Server
Customer	X	
Sender		X
Receiver	X	

Examples for services with this traffic profile are IP television (IPTV), Video-on-Demand (VoD) or Web radio. These are soft real-time applications, thus being delay and jitter sensitive, as well as loss sensitive. Therefore, for this traffic profile the downstream packet stream needs to be audited. The control information from the client to the server is relatively small and insensitive against loss and delay, thus does not need to be audited.

Server streaming traffic profiles could be initiated by the client or the server. The role occupations in both cases are depicted in Table 6 and Table 7, respectively.

Table 7: Roles in a Server-initiated Server Streaming Traffic Profile Scenario

	Client	Server
Customer		X
Sender		X
Receiver	X	

3.2.4 Bulk Data Transfer Traffic Profile

The fourth scenario is the bulk data transfer. Bulk data is simply huge amounts of data. There is a smooth transition between the scenario show in Section 3.2.2 and this one; even the requirements are different. Critical for applications with a client-server interactive traffic profile is the round-trip delay. The throughput is of less importance, because data transferred in each step is typically small - in the range of some kilobytes up to several megabytes. For bulk data transfer, delay is less important, but it requires a high throughput.

Therefore, the transition between client-server interactive and bulk data transfer gets less sharp, when client requests or server responses grow in size, up to a point where traffic characteristics of bulk data transfer obviously prevail. Figure 4 illustrates this fact. It shows an example where a client request data from a server. It is assumed, that the size of the request is small, thus serialization delay is negligible. On the client side two time periods can be identified: The time from sending of the request until the first bit of the response arrives at the client (called t_{RTT}) and the time it takes from the arrival of the first bit until the last bit of the response is received (called $t_{Serialize}$). It is obvious, the bigger the transferred data and the longer the time needed to serialize it ($t_{Serialize}$), the less significant the influence of t_{RTT} is. In the figure, it is also assumed, that jitter and the processing time at the server are negligible.

In scenarios with bulk data transfer traffic profile auditing the available bandwidth between server and client is mandatory, however, may not be sufficient. If the transfer of the bulk data uses the TCP protocol, also the round-trip time is of interest. The TCP protocol uses a flow control mechanism, where data sent needs to be acknowledged by the receiver, in

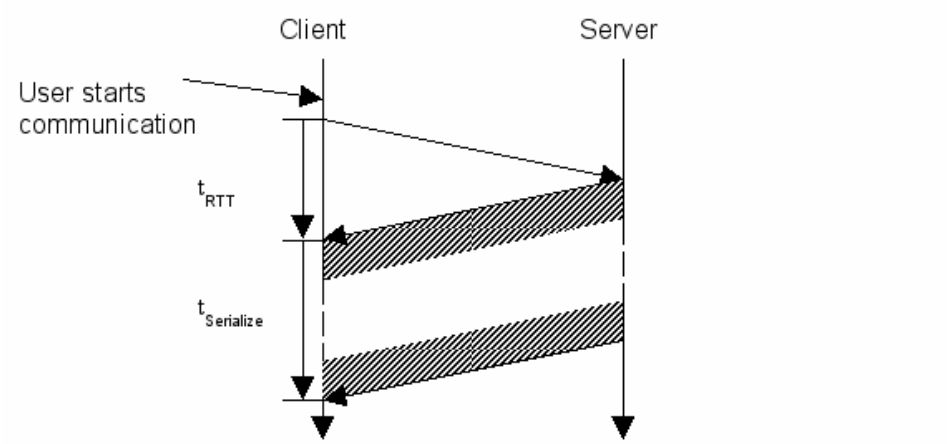


Figure 4: Bulk Transfer Application Scenario

order for the sender to send the next block of data. If the delay is too long, the acknowledgements arrive late and the sender is not able to use the full available bandwidth.

Table 8: Roles in a Client-initiated Bulk Data Transfer Traffic Profile Scenario

	Client	Server
Customer	X	
Sender		X
Receiver	X	

However, the auditing system designed audits only the throughput from server to client in a bulk data transfer traffic profile scenario. If a respective throughput rate is contracted, the ISP has to fulfill all obligations necessary for the client to achieve this rate. Digging for the detailed causes is not the task of an SLA auditing system.

Following this argumentation the server is always the sender and the client is the receiver (see Table 8 and Table 9). The bulk data transfer could be client-initiated, if the client has an SLA with his ISP, or it could be server-initiated if the server holds an SLA with his ISP.

Table 9: Roles in a Server-initiated Bulk Data Transfer Traffic Profile Scenario

	Client	Server
Customer		X
Sender		X
Receiver	X	

3.3 Protocol Sequences

Having identified the respective roles, the next step is defining a protocol for them to communicate. The MeSA protocol fulfills this task. The principle interactions between the different components of the system are already defined in [33]. Besides performing the measurement of the performance parameters, the MeSA protocol also defines messages for session setup and tear-down: It combines communication between the components of

the auditing system and the delay measurement process in one. Therefore the MeSA protocol supports the whole life cycle of an auditing session.

Such a session is established when a user requests auditing services. A session is always created locally in a component and comprises all information related to one auditing service request stored in the respective component. Because several components participate in the provisioning of the auditing service, one service request will lead to several sessions, on in each participating component.

Four roles are involved in the establishment process: the customer, the sender, the receiver, and the Access Network Provider, as described in Section 3.1. These four roles map to three entities, only, as either the sender or the receiver is also in the role of the customer. This first process, where the auditing sessions are established, is called setup phase. During this phase no measurement is performed. Section 3.3.1 describes this process in detail.

The term “session” in this context denotes the usage of the auditing service as such and is independent of any concrete component. The term “session instance” will be used if information stored on a particular component, related to a session, is meant. In this respect, the setup process establishes the auditing session, by creating session instances in all relevant components.

After the sessions are established the auditing service monitors the compliance with an SLA. During this data transfer phase the measurement aspect of the MeSA protocol is most dominant. Auditing reports, which were generated by the receiver, could be sent to the Access Network Providers complaint handler right away, or stored until the session ends. This phase is analyzed and described in Section 3.3.4.

A third phase, called tear-down phase, starts when the auditing session should be terminated. Any of the end hosts — sender or receiver — can request session termination. Besides removing the session instance from all participating components, during the tear-down phase all remaining auditing reports are sent to the customer. Because the message sequences of the tear-down phase relate closely to the ones of the setup phase, this section is described in Section 3.3.3, prior to the description of the data transfer phase. Section 3.3.5 shows how setup and tear-down of a session looks like for four traffic profiles defined in Section 3.2. Building blocks described for the setup and tear-down phase are assembled depending on the detailed role allocations and communication relationships.

Before designing the course of messages in the different phases some preliminary considerations are necessary concerning the storage of state information in the various involved systems. Especially in the Internet with its enormous number of hosts and concurrent connections, scalability issues have to be taken into account from the very beginning of the design process.

3.3.1 Scalability Considerations

Systems participating in providing the auditing service, are the end hosts, i.e., sender and receiver, the Authorization Authority (AA) of the Access Network Provider, and all measurement agents in the border routers. Storing state of user sessions in intermediate systems, *e.g.*, the measurement agents could be a scalability issue, as the available memory in these systems would limit the number of flows possible. Also, the state in the intermediate systems needs to be discarded after the session has ended, which has to happen also in case of failure, *e.g.*, when the discard message is lost.

If measurement agents should be configured for each auditing session separately, another open challenge is determining the measurement agents which need to be set up. Setting up all measurement agents of the whole network is not possible. Setting up only those measurement agents along one route between sender and receiver is also problematic, as because of routing changes the packets of one flow might take a different way through the network. In that case the packets might arrive at a measurement agent, which is not prepared for these packets.

One possibility to omit this problem would be forbidding of routing changes; another one to terminate and restart a session, whenever the route through the network has changed. Both approaches are not feasible: In case of an overload situation or when a link fails, different routes have to be taken. Terminating and restarting sessions would induce a high overhead and requiring a mechanism for the network signaling routing changes to the auditing system. Thus, storing state about auditing sessions is no option and must be avoided.

This argumentation, however, does not apply to the Authorization Authority. This component needs to be placed at the edge of the network, close to the point where the customer network is connected to the ISP network, in order to police service requests. By definition, this requires all auditing messages to pass through the Authorization Authority (or its respective Access Router) independent of any routing changes. Furthermore, systems being placed close to the network edge regularly do not handle that much flow, thus scalability is here less an issue.

The correct termination of sessions is still an issue — but not a new one. A principle called softstate was already proposed in 1984 for Internet protocols, which need to store state in intermediate systems [29]. The principle is conforming with the early design principles of the Internet and is since then widely adopted. Examples are the Resource Reservation Protocol (RSVP) and the Protocol Independent Multicast-Sparse Mode (PIM-SM).

This principle describes a time-out mechanism for the state information. All information stored in the intermediate systems is annotated with a timeout. If this timeout is reached the system removes the information from memory. To avoid the information being discarded, an end system needs to send a refresh message periodically, which retriggers the timeout. Which messages are regards as “refresh messages” is implementation specific.

3.3.2 Phase 1 “Setup”

An auditing session is created, when a user requests auditing services. A prerequisite for requesting these services is an operational SLA with an ISP. The aim of the setup phase is authenticating and authorizing the user, i.e., checking, if he is allowed to use the auditing service. During the setup phase in all relevant systems session instances are created. The selection of these systems has been described above.

The detailed course of messages depends on the traffic profile and the role allocations of one scenario. *E.g.*, in a scenario with a server streaming traffic profile different messages are exchanged depending on whether the sender or the receiver is the customer.

The setup phase implements a request-response protocol consisting of three different messages:

- Init-Req: The message is sent by the customer in order to establish a new auditing session.
- Init-Resp: The message is sent by the opposite end host back to the customer signaling successful establishment of the auditing session.

- Init-Error: This message is sent either by sender or receiver in case of error during the setup process.

In the following subsections the setup phases for different role allocations are analyzed. Later on, in Section 3.3.5, the message sequence are refined and mapped to the four different traffic profiles, which were identified in Section 3.2.

3.3.2.1 Sender-initiated Setup

Depending on whether an auditing session is initiated by the sender or the receiver, the setup phase looks different. If the sender should start an auditing session, it creates a local session instance and then sends an Init-Req message to the receiver. The message is intercepted by the Authorization Authority, which hereupon authorizes the user. Most likely the Authorization Authority will rely on services of other systems to perform this task, *e.g.*, RADIUS or DIAMETER server. Authorization is based on the sender's IP address and the sender's user profile.

On success, a session identifier (SID) and a session instance is generated. The SID has to be unique for the Authorization Authority and for the lifetime of a session. The access router stores the SID in a session table, together with the sender's IP address and the FlowId. The SID is also put in the SID field of the Init-Req message prior to forwarding it. If the authentication fails the router discards Init-Req message and returns an Init-Error message back to the sender containing the reason of the failure.

If the Init-Req message arrives at the receiver, he checks if it is possible to establish an auditing session. Reasons for a failure include different MeSA versions, security concerns, or software failures. If the receiver is not willing or able to start the auditing session, it silently discards the Init-Req packet or returns an Init-Error indicating the reason for the failure.

If the session could be successfully establish the receiver creates a session instance locally and returns an Init-Resp message. A message sequence of a successful sender-initiated setup is shown in Figure 5.

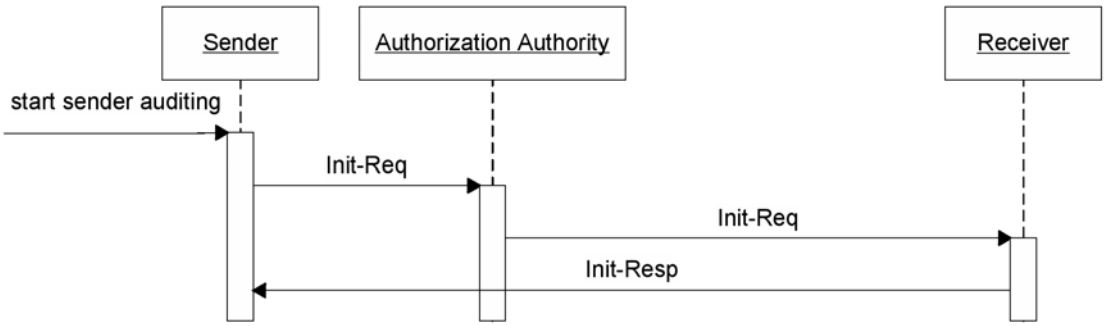


Figure 5: Successful Session Set up (Sender-initiated)

The sender masks packet loss with a retransmission of the Init-Req message: If within a timeout no Init-Resp or Init-Error message arrives, the Init-Req message is sent again. All participating systems must be able to recognize duplicate messages.

3.3.2.2 Receiver-initiated Setup

In a receiver-initiated scenario the customer sends an Init-Req message to its counterpart, too. However, this time the receiver is in the role of the customer, which sends the message to the sender. Figure 6 shows a UML sequence chart of a successful receiver-initiated session establishment.

In a receiver-initiated scenario a session request is authenticated and authorized by the Access Network Provider of the receiver. However, during the data transfer phase the MeSA probes will be sent by the sender and the Access Network Provider of the sender needs to authorize sending of the probes. Therefore a session instance is created in the Authorization Authority of the Access Network Provider of the sender and another one in the Authorization Authority of the Access Network Provider of the receiver. The later Authorization Authority appends information about the successful authentication and authorization to the Init-Req.

Each entity might return an Init-Error message if session establishment is not possible. The message should contain an appropriate error code pointing to the reason why the setup failed. See above for a discussion of this message.

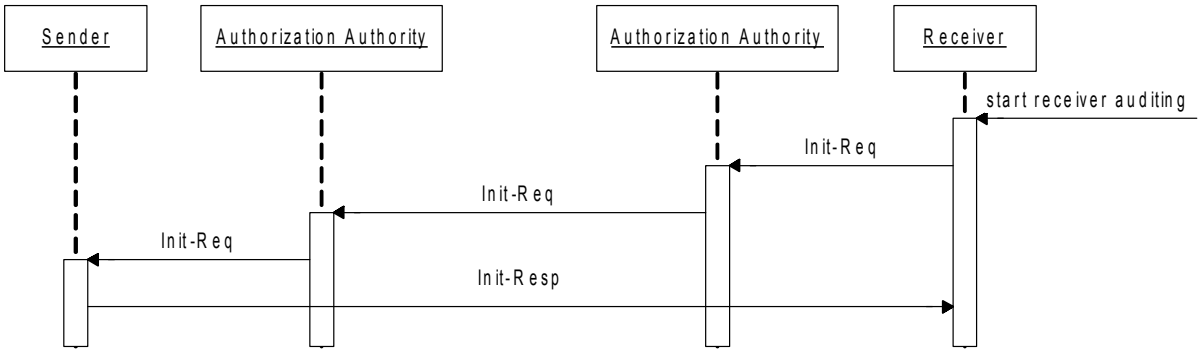


Figure 6: Successful Session Set up (Receiver-initiated)

3.3.3 Phase 3 Tear-down and Result Transfer

The tear-down phase consists of two parts. First, every entity, i.e., the sender, receiver, and the Authorization Authority or Authorization Authorities, is informed about the termination of the session. This triggers the second part, where remaining auditing reports are sent to the entity being in the role of the customer. The process and the exchange of messages depends on one hand whether the sender or the receiver initiated the auditing session, and on the other hand who of both terminates the session. The processes for each of those four cases are described in the following sections:

- Section 3.3.3.1: Sender is customer and terminates the session.
- Section 3.3.3.2: Sender is customer, but receiver terminates the session.
- Section 3.3.3.3: Receiver is customer and terminates the session.
- Section 3.3.3.4: Receiver is customer, but sender terminates the session.

Four messages are defined for the first part of this phase:

- Term-Req: This message is sent by the customer to the opposite end-host requesting a session tear-down.
- Term-Resp: This message acknowledges the reception of a Term-Req message and concludes the first part of the tear-down phase.
- Term-Prep: If the customer and the one terminating the session are two different entities, this message informs the customer that the auditing session should be terminated. The message is only used in this case.
- Term-Error: If an entity receives a termination message, which is not according to the defined message course, or if for any other reason the session could not be terminated, the entity can return a Term-Error message.

3.3.3.1 Sender is Customer and Terminates the Session

The following section describes the case when the sender is customer and later on terminates the session. The termination of an auditing session is always triggered exogenously to the auditing system, i.e., by the user stopping auditing manually, or by the application stopping auditing after all data is sent.

In order to stop auditing the sender sends a Term-Req message to the receiver (see Figure 7). The message is intercepted by the access router and forwarded to the Authorization Authority. It hereupon destroys the local session instance and removes all corresponding entries from its session table. Further, the accounting record is closed and the message is forwarded. After this step the access router will not anymore forward MeSA probes with this session's ID.

Upon reception of a Term-Req message the receiver sends all remaining auditing reports to the sender. In this case only the sender, being in the role of the customer, is allowed to complain about SLA violations to his ISP.

After sending all auditing reports the receiver acknowledges the Term-Req message with a Term-Resp message and destroys its session instance. The process is finished after the Term-Resp message arrives at the sender.

The sender repeats sending of a Term-Req messages for several times, if within a timeout interval no Term-Resp message is received. If the Authorization Authority could not find an entry for the session named in the Term-Req message in its session table a Term-Req retransmission is assumed and the packet is forwarded without further actions. The receiver behaves analogously.

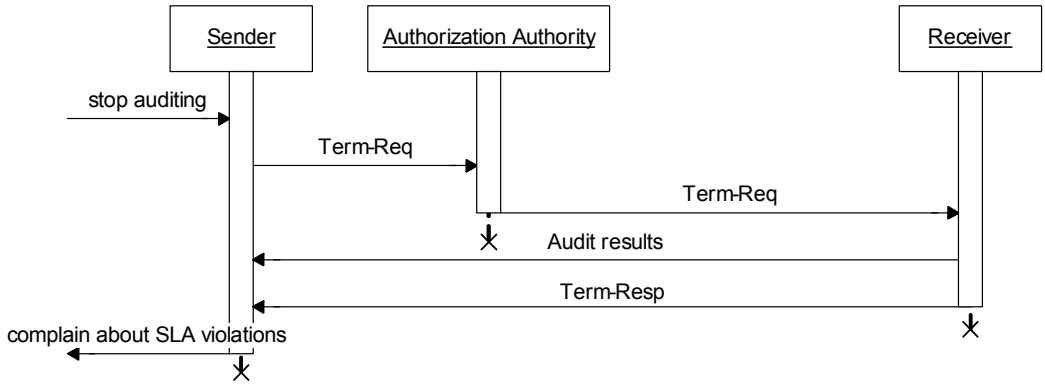


Figure 7: Auditing Session Termination (Sender is Customer and Terminates the Session)

3.3.3.2 Sender is Customer, but Receiver Terminates the Session

If the receiver wants to terminate the auditing session, the process looks a bit different (cf. Figure 8) compared to one shown in the previous section. The receiver, because he is not in the role of the customer, needs to announce the session termination wish with a Term-Prep message to the sender. The sender then starts the same process as depicted in the previous section. After sending of the Term-Prep message, the receiver waits for the Term-Req message some time. If the message does not arrive within a timeout interval, packet loss of either the Term-Prep message or the Term-Req message is assumed and the Term-Prep message is sent again.

The announcement of the termination is necessary because the receiver is not allowed to terminate an auditing session which was established by the sender. For security reasons only the sender, being in the role of the customer, is allowed to do this. If the sender

receives an invalid Term-Prep message, he can return a Term-Error message or silently discard the Term-Prep message. This action, however, will cause the receiver to send the Term-Prep message again.

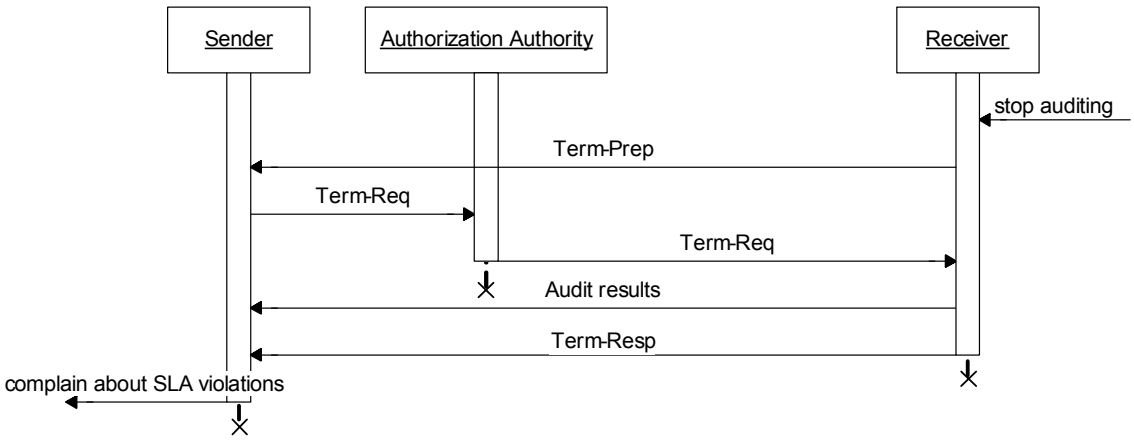


Figure 8: Auditing Session Termination (Sender is Customer, but Receiver Terminates the Session)

3.3.3.3 Receiver is Customer and Terminates the Session

The next two sections analyze termination of an auditing session, which was set up by the receiver, i.e., the case when the receiver is in the role of the customer. The significant differences to previous cases shown before, are the presence of a second Authorization Authority, but therefore it is not necessary sending the auditing reports to the other end host.

Figure 9 shows the process of termination if the receiver is in the role of the customer and later on terminates the session. The Term-Req message takes the same course as the Init-Req message before. It is triggered exogenously to the auditing system and sent from the receiver to the sender. The message is intercepted by the access router of the Access Network Provider of the receiver. The ISP removes all corresponding session instances and forwards the message afterwards.

The Authorization Authority of the Access Network Provider of the sender intercepts the message, too. It destroys all session instances and forwards the message to the sender. The sender, too, deletes all references from its respective table. Further on, no MeSA probes for this session are generated by the sender anymore. The sender acknowledges the Term-Req message with a Term Resp message.

No auditing reports need to be sent during the session or on its termination, as the receiver, who is also in the role of the customer, evaluates the auditing results himself. Like in the previous cases the receiver masks loss of messages by retransmission. All systems, i.e., both Authorization Authorities and the sender, must be aware of handling duplicate messages either by simply forwarding them or answering to them with a Term-Resp message, respectively.

3.3.3.4 Receiver is Customer, but Sender Terminates the Session

In the last of the four possible cases for the termination phase the receiver had initiated the auditing session, i.e., is in the role of the customer, but the sender terminates the session. A Term-Prep message is used by the sender to announce the termination wish to the receiver. The same argumentation holds as for the Term-Prep message above: because of

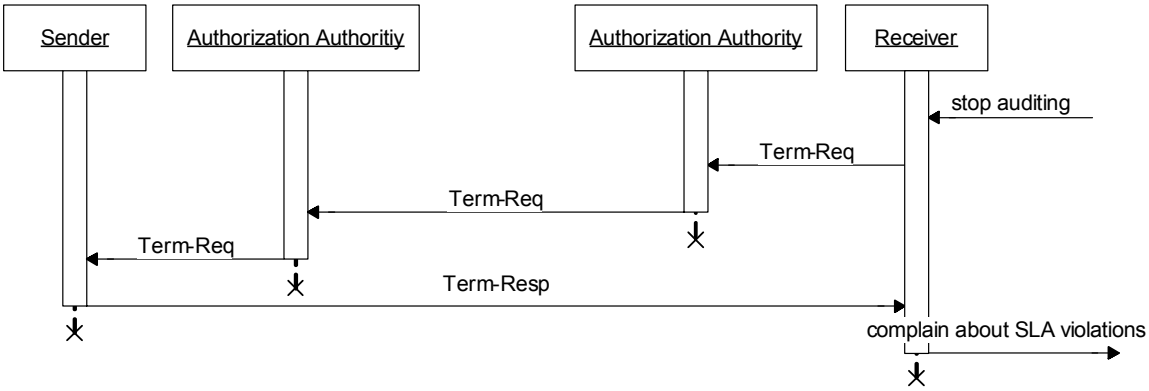


Figure 9: Auditing Session Termination (Receiver is Customer and Terminates the Session)

security concerns the sender is not allowed to terminate the auditing session, which was created by the other end host.

The Term-Prep message starts a process analog to the process described in the previous section. The receiver sends a Term-Req message, which removes the auditing session from the Authorization Authority and the sender. The sender stops generating MeSA probes and answers with a Term-Resp message (see Figure 10). The sender resents the Term-Prep message for a defined number of times, if no answer, i.e., a Term-Req message, returns within a timeout interval. Analogously, the receiver repeats sending of Term-Req messages if no Term-Resp message is receive within a timeout interval.

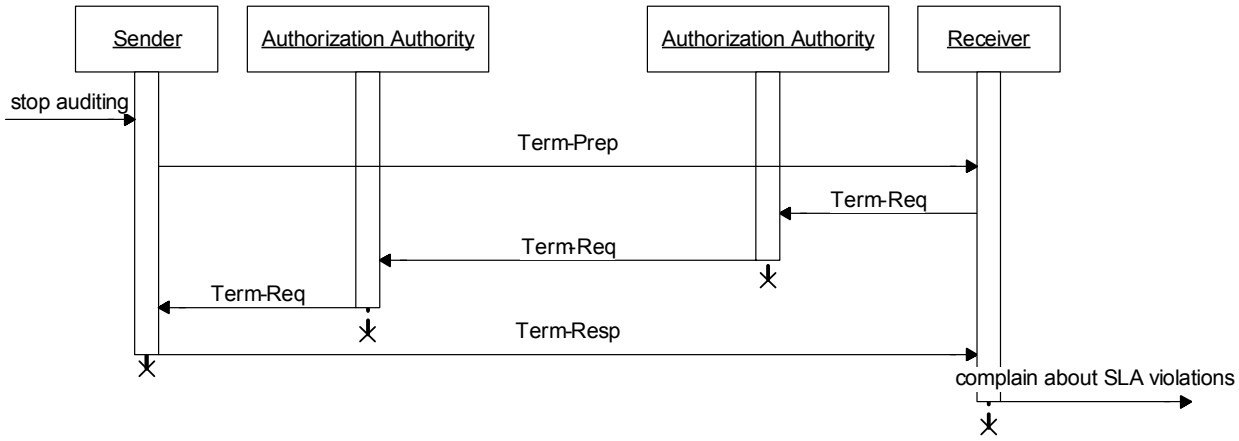


Figure 10: Auditing Session Termination (Receiver is Customer, but Sender Terminates the Session)

3.3.4 Data Transfer Phase

The generation of measurement data, i.e. the actual data for the auditing process, happens during the data transfer phase. In this phase, the sender periodically intersperses MeSA packets of type Probe, so called MeSA probes, in the application packet stream.

Measurement is necessary at all domain boundaries in order to localize an SLA violation. Therefore measurement agents need to be deployed on all border routers. Because these routers must support the MeSA protocol they will be called MeSA enabled routers, or simply MeSA routers.

The Authorization Authority in the sender's Access Network Provider performs access control for each MeSA probe. MeSA probes are only forwarded if they belong to a currently

running session, i.e., a session which was set up successfully and was not yet terminated. If no relating session could be found the packet is discarded.

In order to maintain the softstate of the sessions (see Section 3.3.1), for each MeSA probe the session instance's timestamp is updated. If the timestamp of one session instance is older than the timeout threshold, it is removed.

If a MeSA router receives a MeSA probe, it appends the current time and its own identification to the probe before forwarding it. From the list of timestamps in the MeSA probe the receiver can calculate the delay between each two MeSA routers.

Figure 11 shows a sample scenario, where two end-systems are connected via two providers (ISP A and ISP B). These ISPs operate MeSA routers at their borders. Besides application data and their respective packets, the sender periodically generates MeSA probes. MeSA routers append a new MeSA entry, containing the current time and an identifier of the router to these probes, which is depicted in the lower part of the figure. A probe grows with each MeSA router it traverses by one entry. The maximum size of a MeSA entry is 30 bytes. An IP packet with 1500 bytes can hold up to 48 of these entries. Given that an Internet packet on average only traverses approximately 16 routers at all, this should be sufficient for any healthy Internet path without need for fragmentation. Therefore fragmentation of MeSA probes is not supported.

Finally, the receiver evaluates these probes and calculates the time a probe spent in each domain by subtracting the timestamps. Furthermore, the receiver counts the packets arriving per time interval in order to calculate the throughput.

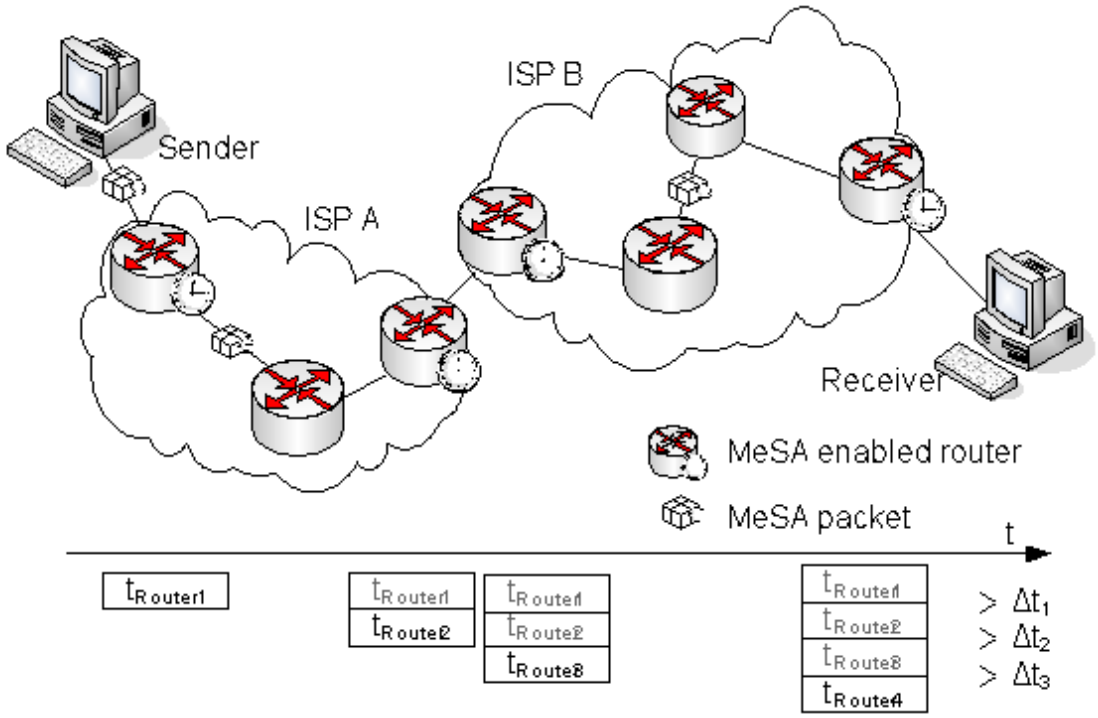


Figure 11: Measuring Delay with MeSA Probes

Sender and receiver in the above example are end hosts, but they could be also special devices, attached to the network and generate and filter out the MeSA probes for a whole network. In this case, however, the advantage of the system to communicate with applications and adapt the measurement process dynamically to the applications' needs would be lost. There are use cases, however, where this is a valid approach. An example is the monitoring of a VPN connection between two sites of an enterprise.

Furthermore, it can be noticed in the figure that the sender does not include a timestamp in the probe, and thus no delay can be calculated for the connection between him and the first router. The reason is the assumption that the connection to ISP A is in the responsibility of the sender for which no SLA exists. Same applies for the connection from ISP B to the receiver.

3.3.5 Setup and Tear-down for Different Traffic Profiles

Depending on the traffic profile of the application and the role allocations, respectively, functionality of the auditing system are performed by different entities. Thus communication relationships and information exchanges vary between different traffic profiles. Invariant building blocks of messages exchanged during setup and tear-down have been identified and described in the previous sections.

This section assembles and refines the building blocks and shows for each traffic profile how the three phases of the MeSA protocol actually look like.

3.3.5.1 Peer-to-peer Traffic Profile

The peer-to-peer traffic profile was first described in Section 3.2.1. Shortly repeated, it is characterized by two almost identical data flows in opposite directions. Each of the unidirectional flows needs to be audit separately. Two cases had been identified:

- Caller-initiated: if the caller is the customer
- Callee-initiated: if the callee is the customer

Figure 12 sketches the course of a caller-initiated peer-to-peer traffic profile session. The three phases (setup phase, data transfer phase, and tear-down phase) of the lifetime of an auditing session are marked on the right-hand side of the figure. Please note that in a peer-to-peer traffic profile scenario two independent, but coupled auditing sessions, one for each direction, are created.

During the setup phase the session with its various session instances are established. In a caller-initiated scenario, the caller sends a request to the callee, who returns a response message in order to signal success or failure. Because in a peer-to-peer traffic profile scenario each end host is sender and receiver at the same time, both Authorization Authorities - the one of the Access Network Provider of the sender and the one of the Access Network Provider of the receiver - must create an auditing session instance. Therefore in the case of a peer-to-peer traffic profile scenario the more complex receiver-initiated session setup is used (see Section 3.3.2), irrespectively if caller or callee is customer.

During this setup process the Init-Req message is read by the Authorization Authority of the Access Network Provider of the caller, which authenticates and authorizes the session request. Proof of a successful session instance set up on the Authorization Authority, and therefore for a successful authentication and authorization of the service request, is appended to the Init-Req message.

The Access Network Provider of the callee will read the message, too, and create an own auditing session instance. Without this session the Access Network Provider of the callee would not allow sending of MeSA probes, because they would not belong to an established session.

On success, during the data transfer phase both entities send data to each other and include MeSA probes in the packet stream. The respective receiver evaluates the probes and creates auditing reports as needed.

The tear-down phase is started either by the caller or the callee. The process is analogous to the ones shown in Section 3.3.3.1 and Section 3.3.3.2, respectively. Figure 12 shows the case when the caller terminates the session. If the callee wants to terminate the session, he needs to send a Term-Prep message first. Because the callee is in the role of a receiver, too, during tear-down he needs to send remaining auditing reports to the caller.

In this document only the phases of the MeSA protocol are described. Most probably the communication of the application which sends the audited data is also structured in different phases. Setup and tear-down from the point of view of the application (*e.g.*, SIP-based (Session Initialization Protocol) session establishment for VoIP) is not looked at. It happens before the MeSA setup phase started, runs in parallel to it, or is part of the communication phase.

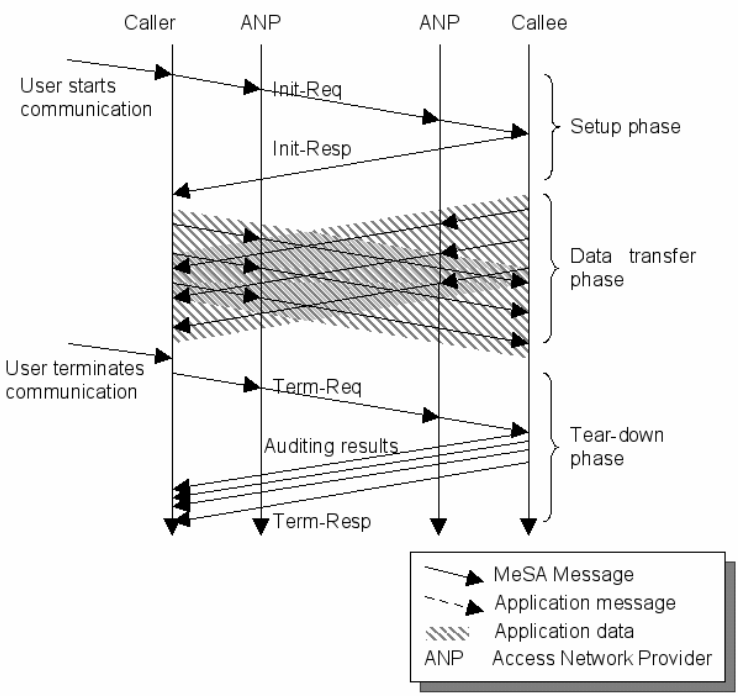


Figure 12: Caller-initiated Peer-to-peer Traffic Profile

In the other case, if the callee is the customer, the sequence chart looks slightly different. The direction of request and response messages is turned around. The caller starts the communication and contacts the callee. This communication step is represented by a dashed line in Figure 13. It is not a MeSA message, but a message of the application, which signals the call. The message is only show in order to demonstrate a consistent sequence: the call is initiated by the caller; however, the auditing is initiated by the callee. Subsequent messages of the application are not shown.

As one response to the arriving of a new call, the callee request auditing by sending an Init-Req message to the caller. Analogously to the previous case the message is read by the Authorization Authority of his Access Network Provider and the Authorization Authority of the Access Network Provider of the caller. The caller acknowledges session setup with an Init-Resp message.

This second example is terminated by the caller, too. Because the caller is not in the role of the customer, he first sends a termination announcement message (Term-Prep message) to the callee, which triggers the termination request-response protocol (see Section 3.3.3.4). This time the remaining auditing results must be sent from the caller to the callee, as the callee is the customer.

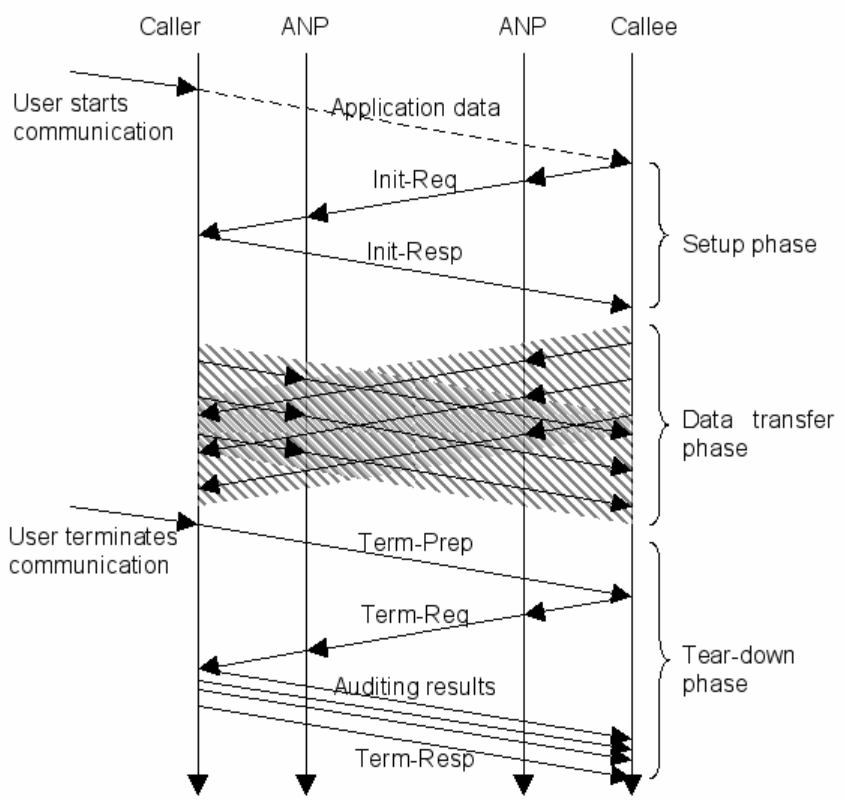


Figure 13: Callee-initiated Peer-to-peer Traffic Profile

3.3.5.2 Client/Server Interactive Traffic Profile

The Client-Server-interactive traffic profile was first introduced in Section 3.2.2. Requests from a client are sent to a server which answers with a response. For the client the time until the response arrives back is the most critical performance parameter.

Figure 14 shows the life cycle of a client-initiated scenario. The user, holding in this example the roles customer, sender, and receiver, starts the communication. A session request is sent by the client, which creates an auditing session instance in the Authorization Authority. The server acknowledges the setup request by sending an Init-Resp message back to the client. The server can keep track of running auditing session, but is not required to do so:

If the server works statelessly, i.e., does not maintain a list of currently running auditing session, it is not able to check if a probe belongs to such a session and it has to reply to all MeSA probes. On the contrary, keeping track of running session enables the server checking if MeSA probes belong to a session or not. MeSA probes not belonging to a session might be silently discarded and not answered.

The server might return the probes immediately, as the auditing system only monitors the performance of the network. The time it takes the server to send the probe is uncritical, as the time the probes stay in the server will be removed during evaluation of the timestamps, anyway.

This means that the time it takes the server processing the request and sending a response is not measured by the auditing system. This time relates to the application performance of the server and is therefore not in the concern of an auditing system for IP carrying. However, in a future extension of the auditing system, the agent at the server could be extended and the processing time in the server could be evaluated.

Finally, when the MeSA probes arrive at the client, they are evaluated and possible SLA violations are detected. During the tear-down phase the auditing session is terminated. A termination request and a respective response message are exchanged, in order to destroy the session instances in the Authorization Authority and in the server. No auditing results need to be exchanged as the client was customer and receiver in one.

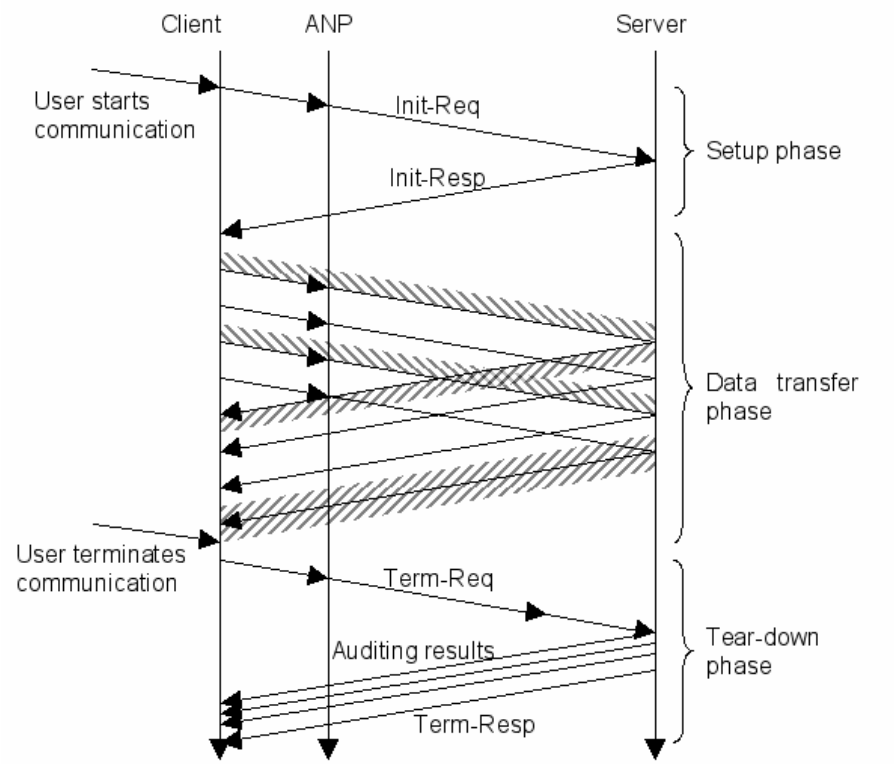


Figure 14: Client-initiated Client-server-interactive Traffic Profile

The case of a server-initiated client-server-interactive traffic profile scenario is depicted in Figure 15. The user starts an application on the client, which requires services from a server. The application will therefore send a message to the server requesting the service, e.g., a Web page from a Web server. This first message sent from the client to the server is not part of the MeSA protocol, as the client probably does not even know that the server wants to audit the application session. However, this message triggers the server to start an auditing session: An Init-Req message is sent to the client. The Init-Req message is intercepted by the Authorization Authority of the Access Network Provider of the server. On successful authentication and authorization of the auditing request the ISP forwards the message. The Access Network Provider of the client intercepts the message, too, and creates a session instance locally. The client finally acknowledges the Init-Req message with an Init-Resp message, which is sent from the client to the server.

The figure depicts the termination process when started from the client, which is the more complex case, as the client is not in the role of the customer. It sends a Term-Prep message to the server, showing its wish to terminate the session. The server sends a Term-Req message which is read by the Authorization Authorities of the Access Network Provider of the server and receiver, respectively. Those remove their respective session instances from memory.

On reception of a Term-Req message the client forwards all remaining auditing results to the server, closes its session instance and acknowledges the termination with a Term-Resp message.

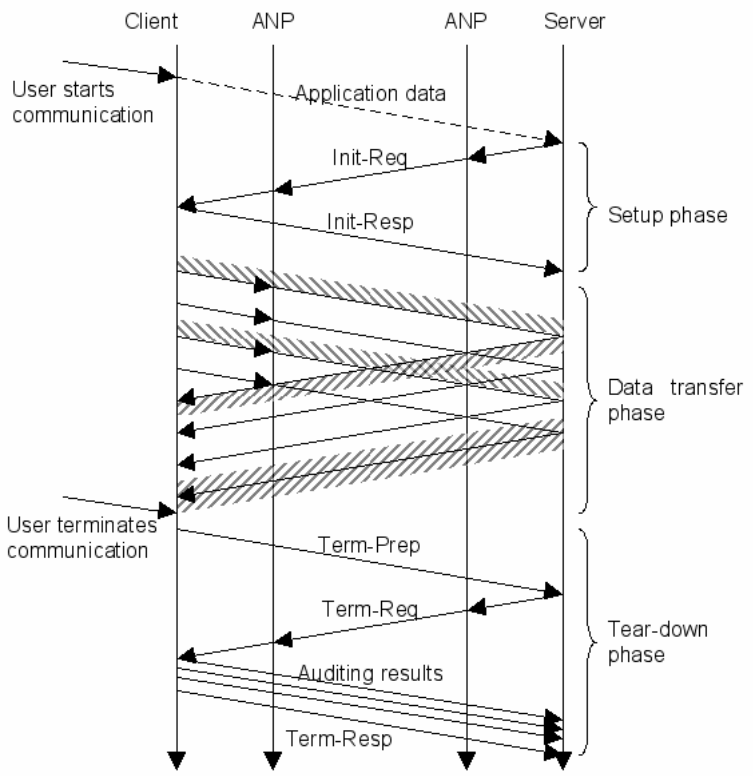


Figure 15: Server-initiated Client-server-interactive Traffic Profile

3.3.5.3 Server Streaming

The third traffic profile, which was identified and defined in Section 3.2.3, is called Server streaming. This traffic profile describes scenarios with one unidirectional continuous data stream from a server to a client and none or almost none traffic in the opposite direction.

In this scenario only the traffic from server to client is audited. The data sent in the opposite direction is not audited, because applications showing this traffic profile only sent control data to the server, which is less quality demanding.

A client-initiated server-streaming traffic profile scenario is shown in Figure 16. Because application data is sent from the server to the client, but the client requests auditing, this case corresponds to the generic receiver-initiated one shown in Section 3.3.2.2.

To start a new auditing session, the client sends an Init-Req message to the server. This message is intercepted by the Authorization Authority of the Access Network Provider of the client, which authenticates and authorizes the request. If this process can be accomplished successfully the message is forwarded. The Authorization Authority of the Access Network Provider of the server intercepts the message again. Please refer to the generic case in Section 3.3.2.2 for a detailed description of the interaction between the both Authorization Authorities. Finally, the server acknowledges session setup with an Init-Resp message.

During the data transfer phase the server periodically includes MeSA probes in the data stream until the server or the client tears-down the session. The tear-down process, started by the client, is shown in figure below, too. It is a refinement of the general case shown in Section 3.3.3.3. The client sends a Term-Req message, which is also read by the both Authorization Authority of the Access Network Providers of the client and the sender, respectively. No auditing results need to be transferred, as all auditing results are already stored at the client, which holds the role of the customer in this scenario.

In a server-initiated scenario a slightly different message sequence during setup and tear-down could be observed. Figure 17 shows an example of this case. The user on the client starts an application which requests a data stream from the server. In order to show a logically consistent sequence this request is shown in the figure below even it does not belong to the MeSA protocol.

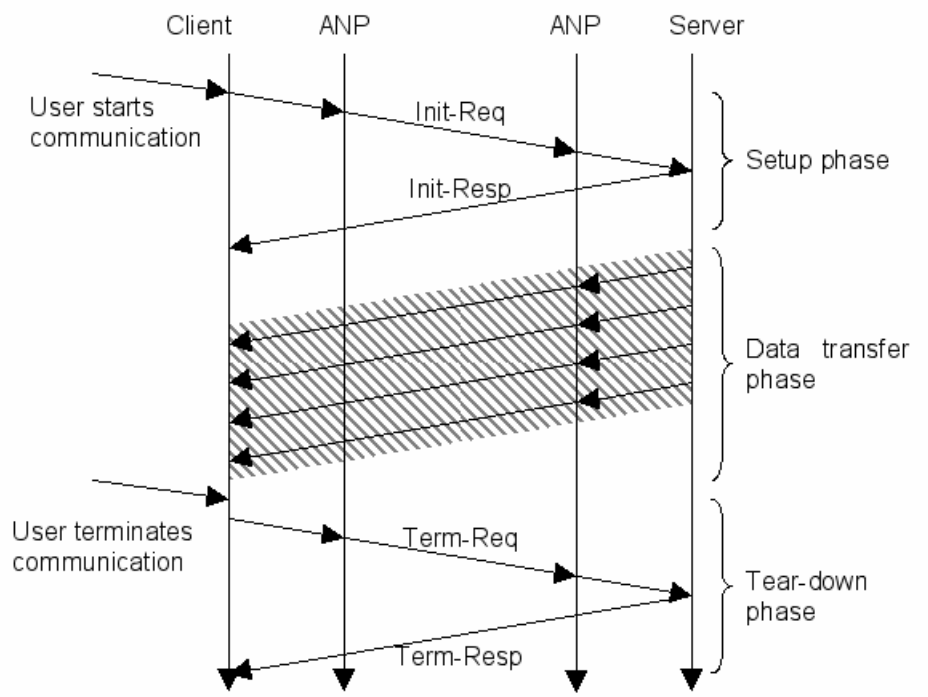


Figure 16: Client-initiated Server Streaming Traffic Profile

The server determines from its configuration that for this request an auditing session should be started. It therefore sends an Init-Req message, which is thereafter authenticated and authorized by the Authorization Authority of its Access Network Provider. On success the message is forwarded to the client who acknowledges the start of the auditing session with an Init-Resp message. The auditing session is fully set up after the server received the Init-Resp message.

During the data transfer phase the server periodically includes MeSA probes in the data stream, which are evaluated by the client. The auditing session could be terminated by the server or the client. In above figure the user terminates the session. The process shown above is based the one shown in Section 3.3.3.2.

Because the user is not in the role of the customer he must ask the customer, in this case the server, to terminate the session. This is done by sending of a Term-Prep message. If the server agrees, it sends a Term-Req message. This message is read by the Authorization Authority of the Access Network Provider of the server which removes all references to the respective session. The client, too, destroys its session instance after it has sent all stored auditing results to server. The tear-down phase is closed by a Term-Resp message sent from the client to the server.

3.3.5.4 Multicast Server Streaming

The MeSA protocol supports IP Multicast in the Server streaming traffic profile scenario, which is the only one, which could be extended to a multicast scenario. MeSA probes are sent to the same address as the application data. Thus, in case of a Multicast IP address, the Multicast routers will copy and forward the probes to the same recipients as the

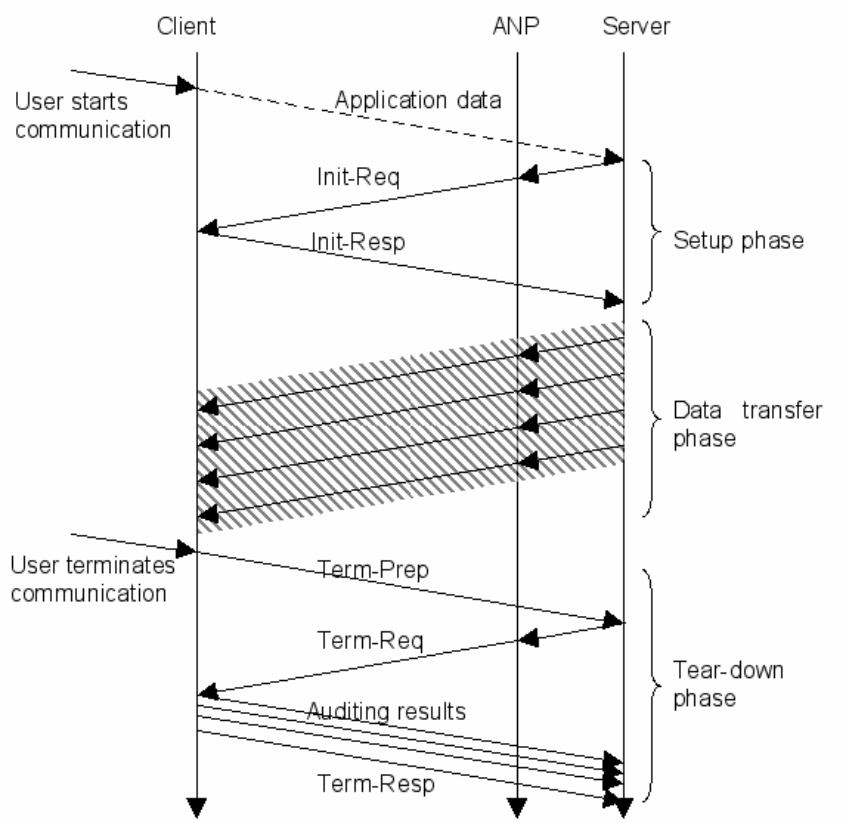


Figure 17: Server-initiated Server Streaming Traffic Profile

application data packets. This case is not listed explicitly in Section 3.2 as from the auditing point of view the multicast server streaming traffic profile does not differ significantly from the (unicast) server streaming traffic profile.

However, the setup and tear-down phases need to be adjusted, as the server could be overloaded by messages sent from the client to the server. Messages from clients to the multicast server are regularly not permitted. Furthermore clients could join and leave multicast groups at any time. Therefore the auditing session instance in the server must be different from the ones in the clients. For the same reason, the notion of server- or client-initiated session is meaningless, as each end hosts starts its session independently.

Figure 18 shows a sample procedure with a multicast server and one client. Before the server starts a transmission it opens a new multicast server auditing session. Therefore it sends an Init-Req message to the multicast address. This message is intercepted by the Authorization Authority of the Access Network Provider of the server. It authenticates and authorizes the request and acknowledges the session set up by returning an Init-Resp message. Opposite to all unicast scenarios, in the present case the Authorization Authority generates the Init-Resp message.

The server then transmits its data to the multicast group. It periodically includes MeSA probes in the data stream. Each probe sent is authorized by the Authorization Authority of the server's Access Network Provider.

At any time clients might join the transmission by joining the multicast group. This process is not part of the MeSA protocol but standard IP multicast procedure. In parallel the client establishes a new multicast client auditing session. The session is created analogously to the creation of the multicast server auditing session: The client sends an Init-Req message to the server, which is intercepted by the Authorization Authority of the client's Access

Network Provider, and authenticated and authorized there. The Authorization Authority acknowledges a successful session setup by returning an Init-Resp message.

Using the server's address for the request is mandatory, even the Init-Req message must never arrive at the server: As the client does not know the address of the Authorization Authority, the server's address is the only one which could be used. But as the Authorization Authority filters out this message and does not forward it, the server is protected from overload situations.

Setting up an auditing session is not mandatory when joining the multicast group. A client might join only the multicast group, but does not request an auditing session. In this case it receives the MeSA probes and is able to evaluate their contents; however, it is not allowed to complain about SLA violations to his Access Network Provider.

Only for times during which a multicast client auditing session was existent the client is allowed to complain about SLA violations. This is not a technical restriction but a business one, which most probably ISPs will enforce. The reason is that the client is likely to be charged a fee for using auditing. This fee is an additional income for the ISP and helps him to improve the network and to compensate for incentives which need to be paid to the customer in case of SLA violations. When the client does not want to receive the data stream anymore, it leaves the multicast group and terminates the auditing session. As described above, maintaining group membership is no task of MeSA. It is shown in the figure only for completeness.

Terminating the auditing session is triggered by the client by sending a Term-Req message. Analogously to the session setup, the message is addressed to the server. The Authorization Authority of the client's Access Network Provider receives the message and closes the session instance. The Term-Req message is not forwarded but a Term-Resp message is sent back to the client, which acknowledges the reception of the Term-Req message and the termination of the session.

The server terminates its auditing session independently of the client. The termination consists of a Term-Req and Term-Resp message exchanged between the server and the Authorization Authority of the Access Network Provider of the server. After the Authorization Authority acknowledges the end of the auditing session with a Term-Resp message, it will not forward any longer MeSA probes of the server.

3.3.5.5 Bulk Data Transfer

The bulk data transfer traffic profile was the fourth and last traffic profile introduced in Section 3.2.4. For bulk data transfer the throughput from the server to the client is the most prominent performance parameter.

Figure 19 shows a sample client-initiated bulk data transfer traffic profile scenario. In the depicted scenario the client is customer and receiver. The server, which sends the bulk data to the client, takes the role of the sender. In this respect the setup phase in this scenario is receiver-initiated as shown in Section 3.3.2.2.

The Init-Req message of the client is authenticated and authorized by the Authorization Authority of the Access Network Provider. On success proof is added to the message, which is evaluated by the Authorization Authority of the server's Access Network Provider. This Authorization Authority creates a session instance, too, and forwards the message to the server. The server acknowledges the Init-Req message with an Init-Resp message.

On request of the client, the server sends the data. It periodically includes MeSA probes in the data stream, which are evaluated by the client. In the figure only one download is

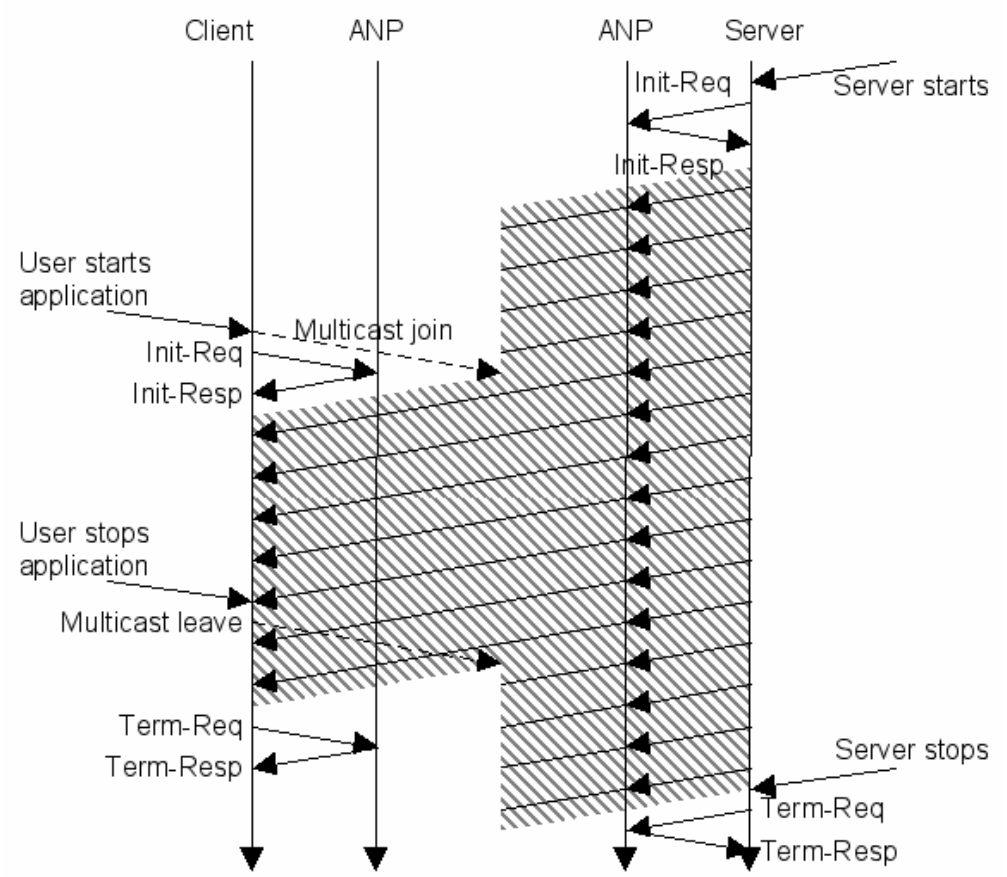


Figure 18: Multicast Server Streaming Traffic Profile

shown, however, it is not necessary to terminate an auditing session after only one interaction. Several downloads are possible during the data transfer phase.

In the figure shown, the auditing session is terminated by the client, i.e. the receiver, who is also the customer (analogous to Section 3.3.3.3). The opposite case where the session is terminated by the server is shown in Section 3.3.3.4.

A Term-Req message is sent by the client, which closes the auditing session in the Authorization Authorities of the Access Network Provider of the client and server, respectively. The server acknowledges the termination of the session with a Term-Resp message.

A server-initiated bulk data transfer traffic profile scenario is shown in Figure 20. When the user requests the bulk data from the server, the server first starts an auditing session. Therefore an Init-Req message is sent, which is authenticated by the Authorization Authority of its Access Network Provider. The client acknowledges the session setup with an Init-Resp message.

After successful session setup the server sends the data to the client. As in the previous examples, during the data transfer phase it periodically includes MeSA probes in the data stream. The MeSA probes are authenticated by its Access Network Provider's Authorization Authority.

The figure also depicts a termination process, which is started by the client. The client in this example is receiver but not customer, therefore the process is analogous to the one shown in Section 3.3.3.2. The tear-down phase for the case where the server terminates the session is shown in Section 3.3.3.1.

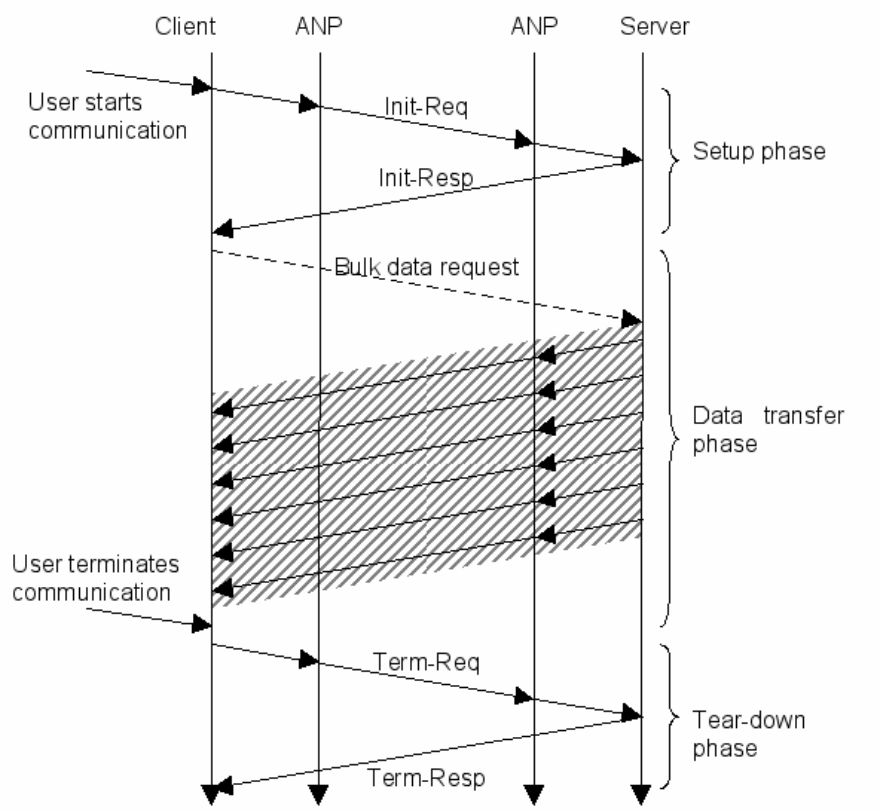


Figure 19: Client-initiated Bulk Data Transfer Traffic Profile

With a Term-Prep message the client asks the customer, i.e., the server in this case, to stop auditing. A Term-Req message starts the actual tear-down process. On reception of this message the client sends all remaining auditing results to the server and closes the session by sending a Term-Resp message.

3.4 Simulations of Multi-domain Auditing

The purpose of the simulations is to verify the proper operation of the MeSA probing technique which is the core component of the ASAM approach; that is to verify the ability of MESA probes to properly depict the QoS treatment of application flows in an end-to-end multi-domain environment. This will allow for proper SLA auditing and detection of SLA violations within each domain.

For simulation purposes a special extension to ns-2 has been developed, which allows for realistic (as derived through measurements performed in the EMANICSLAB) values of jitter to be introduced in network elements. This means that simulations can be carried out without the need to introduce synthetic background traffic in order to introduce jitter, therefore they can be kept simple and lightweight.

The topology which is going to be used for simulation purposes is shown in Figure 21.

In this topology, the multi-domain environment is composed of 3 autonomous systems (ASes) connecting a sender-receiver pair. For the sake of simplicity each AS is abstracted as consisting of a pair of border routers (every router being MeSA capable) connected through a single link.

At each link, a jitter will be introduced using the ns-2 extension developed, which affects both the application traffic between sender and receiver itself and the MeSA probe traffic.

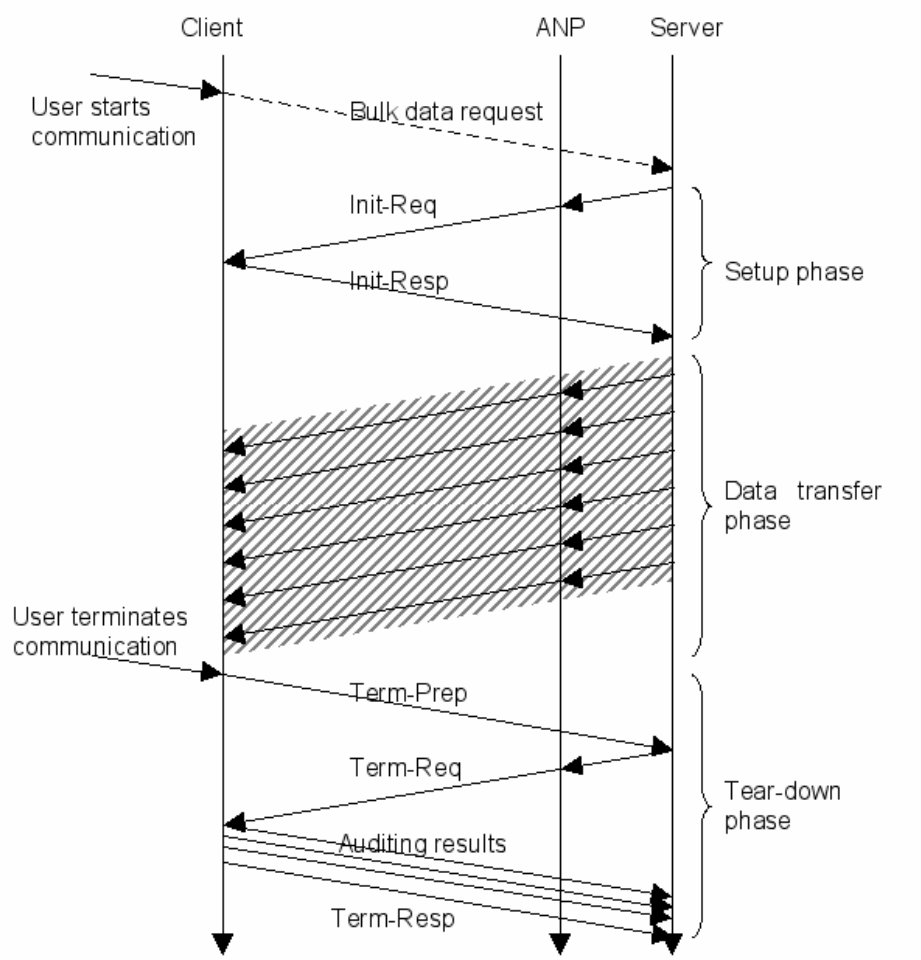


Figure 20: Server-initiated Bulk Data Transfer Traffic Profile

As mentioned already in previous sections, the MeSA probing technique can measure delay and jitter metrics within each AS in a multi-domain chain; therefore delay and jitter are the QoS metrics that will be used in the simulation based evaluation.

The objective of the simulations will be verify whether MeSA probes can successfully depict the delay and jitter experienced by application traffic between sender and receiver with appropriate tuning of the probe characteristics (i.e. periodicity, rate and duration of probes).

Simulations will be performed for application traffic following the traffic profiles described in previous sections and the parameters of the probes will be properly adjusted for each application traffic scenario. The simulation analysis will be performed by post processing the ns-2 generated trace files in order to derive and compare delay and jitter values for application and probe traffic.

Figure 21: Simulation Topology

3.5 Implementation Architecture of SLO Compliance Monitor

Based on the role and protocol description in previous sections, an implementation architecture of the SLO Compliance Monitor (SLO CM) is designed as depicted in Figure 22. The architecture comprises following components:

- Commander: This is the component that controls other components within SLO CM. It also communicates with the SLO CM of the corresponding end system to exchange control messages, *e.g.*, Term-Prep message.
- AA Client: This component communicates with the Authorization Authority of the Access Network Provider based on the signal from the Commander.
- Reporting Server: This component is responsible for collecting audit results (reports) from other SLO CMs.
- Reporting Client: This component sends audit reports to other SLO CMs, if the corresponding end system is the customer (the initiator of the auditing session). Otherwise, the resulted audit reports are placed into the database for further processing.
- MPG Configurator: This component is responsible for configuring MeSA Probe Generator concerning, *e.g.*, type of probe packets and sending frequency.
- MPC Controller: This component receives notification message from MeSA Probe Collector if MeSA probe packets are available. Upon reception of this message, it requests Meter Data Retrieval to get the MeSA probes.
- Meter Data Retrieval: As mentioned, this component retrieves MeSA probe packets collected by MPC and delivers them to the Compliance Evaluator.
- Compliance Evaluator: This component implements the functionality to audit MeSA probes according to SLO agreed upon between the customer and his Access Network Provider. It generates a report if there is a violation to the SLO. Those reports are delivered to the Reporting Client.

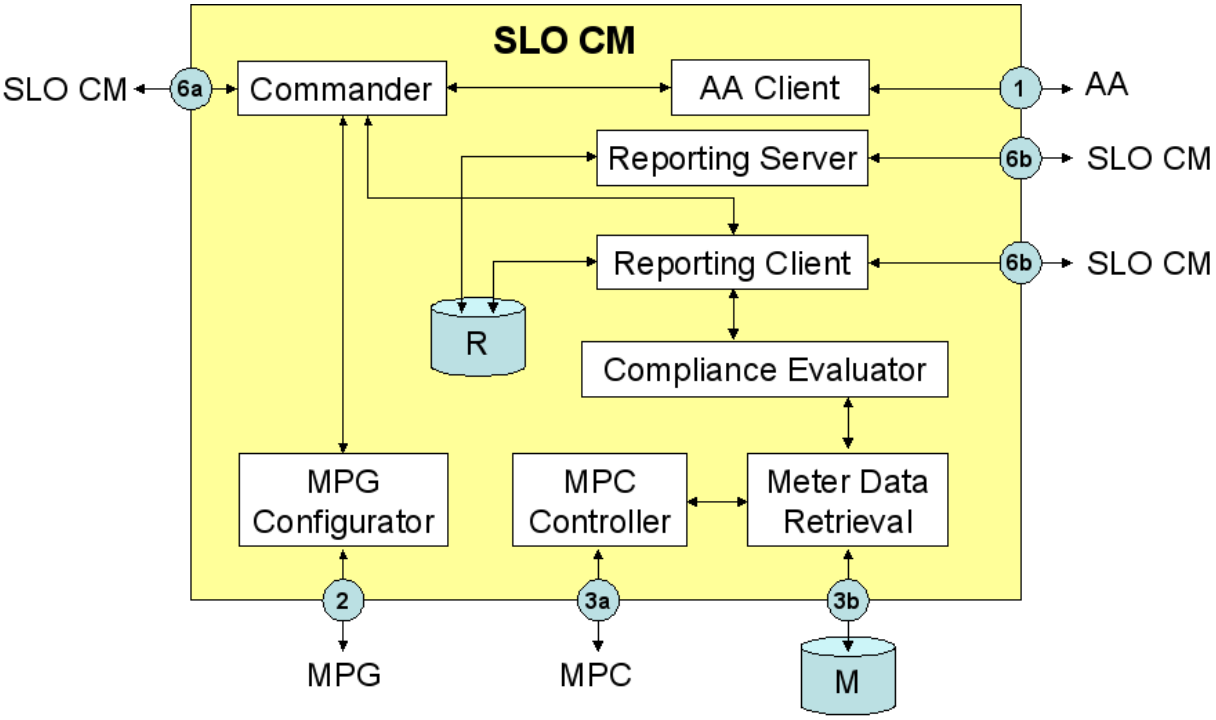


Figure 22: Implementation Architecture of SLO CM

3.6 Evaluations

Customer satisfaction when using services is a key requirement for durable business relationships. Especially in the Internet, where several providers need to cooperate in order to deliver a service to a user, it is necessary to monitor if services are provided as contracted. The ASAM framework provides a means to monitor the quality of IP carrying services and evaluate if they have be provided according to an SLA.

As soon as IP carrying is not a one-time usage product, but a quality product, for which a user is willing to pay more if he receive therefore a better quality, ASAM provides the possibility for the user to evaluate the product's quality and actually see that he receives what he paid for.

But also for the provider it is a possibility to stand out from the crowd and receive higher revenue for a better service. Higher investments in the network only pay if the user is willing to pay higher tariffs for a better service. But as long as it is not possible to measure the “better service” less people will be up to do so.

ASAM provides a rather lightweight framework, which is heavily based on the participation of the communication partners and a bandwidth saving probing approach, and which can evaluate the compliance with an SLA. In the previous sections traffic profiles and respective information exchange requirements have be evaluated. These have been taken to define a protocol which supports the whole life cycle of an auditing session.

A simulation scenario has been depicted, which will be used to asses the capabilities of ASAM in a multi-domain environment. Here especially the ability to determine the location where an SLA violation happened will be tested.

3.7 Complimentary Work

ASAM defines an architecture and a new protocol for metering and auditing of network performance across multiple, co-operating network providers. To address the application of SLO compliance monitoring in a specific scenario for hosted streaming services, a paper titled “Monitoring of SLA Compliances for Hosted Streaming Services” that complements ASAM work has been written [16] (*cf*. Section 18.1). The application scenario selected shows a careful choice of SLA parameters, precise specifications of SLOs, and appropriate reimbursements, in case of violations to those SLOs specified. In this scenario, the Service Provider (SP) offers streaming services and has its streaming server (software) hosted by a Network Provider (NP) and concludes an SLA with it. The streaming server runs on nodes provided and owned by the NP. Key SLA parameters relevant to this service are availability, latency, and bandwidth.

The paper focuses on bandwidth parameter and defines it as the number of bytes transferred between streaming servers and Points of Presence (PoP) of the NP within a pre-defined time interval T. The downlink bandwidth is distinguished from the uplink bandwidth, since a streaming service has different requirements with respect to these two directions. The bandwidth SLO defines in general the minimal ratio between the outgoing and the incoming data rate. Depending on where the incoming and outgoing data rates are measured, the SLO can specify four different levels of granularity of data rate aggregation: not aggregated, aggregated at access routers, aggregated at PoPs, and aggregated at access routers and PoPs.

The architecture for the SLO compliance auditing is designed based on following key requirements for a technically and economically feasible auditing infrastructure: multi-

domains support to allow secure inter-domain interactions, load scalability, flexibility to cope with SLO updates in the change management. The architectural components are distributed and comprise audit manager, metering, compliance auditor, and reimbursement. This paper also defines protocol interfaces for components interactions. To achieve a fair reimbursement scheme, this paper proposes to calculate reimbursement based on degree and duration of SLO violations, since current practices in industry are either based on a fix percentage of monthly charges or on the length of the time required by the provider to restore its service performance. In case of Bandwidth SLO, the degree of a violation is either the maximum or the average deviation of the difference between incoming and outgoing rate from the committed threshold.

To study and show the feasibility of the design, a bandwidth SLO compliance auditor has been implemented prototypically on top of the generic auditing framework AURIC and evaluated with respect to requirements specified. The use of the Diameter protocol for a secure and reliable inter-domain communication depends to a great extent on the OpenDiameter implementation applied (TLS or IPsec for security, and TCP or SCTP for a reliable communication). The prototypical implementation shows a linear scalability of processing time with respect to the number of measurement records, and the analytical evaluation shows that SLO changes can be accommodated easily. To complete the feasibility evaluation, economic gain is studied. An operational automated auditing infrastructure provides economic gain in forms of: (a) a chance to take corrective actions at a very early stage of an SLA violation. This avoids potential greater loss caused by more customer claims, (b) reduced efforts in Customer Relationship Management (CRM), (c) reduced efforts of technical support team. Therefore, automated SLA compliance auditing can potentially achieve an economic gain for a provider and the customer, in particular for a large number of customers and services.

4 SaPDoGS: SLA and Promise Descriptions of Grid Services

SaPDoGS has investigated the use of Promise Theory and the voluntary cooperation paradigm to understand Service Level Agreements (SLA) in organizations generally, with a particular look at Grid services. SLA is a concept now used loosely as well as in contract terms between providers and customers. At a discussion during the International Conference on Large Scale Systems Administration (LISA 2008), a discussion amongst 17 attendees (many from Fortune 500 companies) concluded that the service paradigm for management of IT is simple and useful, but sometimes also simplistically applied. There is presently too much focus of simple SLA slogans like “five nine's reliability” without rational motivation. The SLA is however the only tool people recognize — but SLA is often the wrong term, “service promise” is a better term, because so-called SLAs are not always agreed upon by all parties in all parts of an organization – they are only promises.

For business or economic alignment, user or customer experience is the key driver. One must look for metrics for this. Service level management seems more important than the SLA itself (i.e. the SLA is like a flag to rally the armies of the organization), but the processes it unleashes are mainly in someone's head, not documented — so often the only thing written down is the SLA. Several attendees conversant with promise Theory indicated that it had helped them in deploying IT infrastructure because one of the major challenges in large scale infrastructure is not technical but political, i.e., how to get different departments and groups to cooperate freely in a distributed computing project?

Based on a vast number of SLAs currently used throughout the industry which are contracts in plain natural language documents including a small set of Quality-of-Services (QoS), namely response time, throughput and availability. Other important metrics are never mentioned or are deliberately not mentioned. It is difficult to say which metrics are easy to automate and which metrics are commonly used in different IT processes and services.

Much has been written about the Grid from the viewpoint of architectural engineering, but such high level views obscure the underlying details of Grid interactions.

- Promise theory is a natural framework in which to describe the Grid, because it deals with assemblies of completely independent agents which can then work together by making promises to one another.
- Promises can capture ideas like virtualization easily.
- By starting at a low level with individual autonomy, promise theory becomes a discipline by which to document the necessary and sufficient promises required to enable collaborative work.
- Beyond that, it provides a simple mathematical framework for identifying virtual “organization” in terms of patterns of promises exhibited by agents.

4.1 A Categorization Scheme for Promises in Grid

4.1.1 Service Level Agreement vs. Promises

In order to compare promises and SLAs, several questions and issues have to be clarified. Let us start to describe how SLAs are defined and structured and then what does mean promise theory for Grid users — which influence has this concept on business and economic management. But before starting with Grid and promises and why promises are important for Grid, let us try to answer the question: Is Grid ready for business applications?

Several years ago Grid applications have been involved in research allowing the transparent access to distributed resources, but future Grids aim for the industrial and business market. For also attracting the commercial user important standards such as reliability, transparency, QoS, and Quality-of-Devices (QoD) still need to be officially recognized as major requirements for the implementation of future Grids at a commercial level. Commercial Grid today needs to provide much stricter guarantees. These guarantees have to be specified in terms of promises and have to be monitored and assured.

Lack of experience in the use and automation of performance metrics causes problems for many organizations as they attempt to formulate their SLA strategies and set the metrics needed to support those strategies. Although SLAs are the only tool people recognize and use, SLA means obligation and this term does not satisfy the customer need and doesn't describe precisely the resources the providers are supposed to deliver. Thus effective promises are extremely important to assure business continuity, customer satisfaction and trust. The metrics used to measure and manage performance compliance are the heart of a successful agreement and are a critical long term success factor.

A Service Level Agreement (SLA) is a formally negotiated agreement that spells out the level of service that will be delivered from a service provider to its costumer. In this way it is both a commitment and a score card [26].

According to Shally Bansal Stanley in Network world “the truth is that SLAs are nothing more than insurance policies. Just as life insurance doesn't guarantee life, SLAs don't guarantee levels of services. They provide you with compensation in case something goes

wrong". This is the first promise supposed that two agents have to take into account especially in multiple management cross domains. Promises are either negotiated between two agents. Negotiating an agreement is an exchange (protocol) of messages between agents (customer and provider), potentially involving some form of a middleman or broker or a third party. The result is an agreement of all terms that are important for either side.

When a customer finds a service that he wants to use, first he makes some promises such as he engages in a negotiating phase with various available service providers, after an agreement has been achieved, the customer and the service provider then sign this agreement. This agreement is called SLA. The Tele Management Forum defines an SLA as "a formal negotiated agreement between two parties, sometimes called a service level guarantee. It is a contract (or part of one) that exists between the service provider and the customer, designed to create a common understanding about services, priorities, responsibilities, etc. ..." so an SLA promises what is possible to deliver and deliver what is promised.

In order to use the Grid technology for business purposes, more promises in the Grid technology have to be studied to realize current and future ASP (ASP is a business concept between financially independent entities) business models that integrate distributed and heterogeneous resources.

In Grid environment users must be given some form of commitment and assurances on top of the allocated resources, such as performance, security, availability, latency, throughput, jitter, variance, etc. as well as assurances dealing with erroneous conditions, fail-over policies or backups and more. Those promises need to be agreed upon before use and manifested in form of SLA. What is more, the quality that can be expected by a service must be part of the service location process and the SLA must be in a form to allow automated supervision.

This leads to fulfilling the expectations of all agents and it provides metrics for accurately measuring performance to the guaranteed Service Level Objectives (SLO). The defined metrics will be used to detect violations to the promised SLOs and to derive consequential activities in terms of rights and obligations during the monitoring and analysis phase. They play a key role in metering, accounting and reporting and provide data for further analysis and refinement of SLAs. SLA metrics are defined from a variety of disciplines, such as business process management, service and application management, or traditional systems and network management.

4.1.2 SLA Definition

SLAs can be grouped into different categories [27] according to several purposes such as:

- Basic Agreement: Defines the general framework for the contractual relationship and is the basis for all subsequent SLAs inclusive the severability clause.
- Service Agreement: Subsumes all components which apply to several subordinated SLAs.
- Service Level Agreement: Normal Service Level Agreement
- Operation Level Agreement (OLA): A contract with internal operational partners, which are needed to fulfil a superior SLA.
- Underpinning Contract (UC): A contract with external operational partner, which are needed to fulfil a superior SLA.

The SLA could be an internal or external agreement. As explained in [6], an internal agreement is rather an informal agreement than a legal contract (in-house agreement between internal departments or divisions). An external agreement may be between a service provider and an external service consumer or it may be a multi-tiered agreement including third parties up to a multitude of parties. This leads to distinguish between [6]:

- Standard Agreement: Standard contract without special agreements
- Extensible Agreement: Standard contract with additional specific agreements
- Individual Agreement: Customized, individual agreements
- Flexible Agreement: Mixture of standard and individual contract

A typical SLA [2] has the following components:

- Purpose: Describing the reasons behind the creation of the SLA
- Parties: Describes the parties involved in the SLA and their respective roles.
- Validity Period: Defines the period of time that the SLA will cover. This is delimited by start time and end time of the term.
- Scope: Defines the services covered in the agreement.
- Restrictions: Defines necessary steps to be taken in order for the requested service levels to be provided.
- Service level objectives: Are the levels of service that both the users and the service providers agree on, and usually include a set of service level indicators, like availability, performance and reliability. Each aspect of the service level, such as availability, will have a target level to achieve. Service Level Objectives have daytime constraints associated with them, which delineate their validity.
- Service level indicators: The means by which these levels can be measured. Service Level Indicators (SLI) are the base level indicators.
- Penalties: Spells out what happens in case the service provider underperform and is unable to meet the objectives in the SLA. If the agreement is with an external service provider, the option of terminating the contract in light of unacceptable service levels should be built in.
- Optional services: Provides for any services that are not normally required by the user, but might be required as an exception.
- Exclusions: Specifies what is not covered in the SLA.
- Administration: Describes processes created in an SLA to meet/measure its objectives.

SLAs are currently one of the major research topics in Grid Computing resulting in a language definition for specification of SLAs and corresponding architectures [28]. In fact, a broad number of Grid applications require workflow processing beyond the simple execution of monolithic jobs, because the resources (hardware and software) as well as expertise are distributed over multiple sites. Thus the problem grows more complex in terms of management or SLA specification.

Grid computing emerged as a paradigm of sharing resources for collaboration and resource usage optimization purposes. A Grid is made up of a finite set of nodes, where a node is a system managing a set of resources [5]. A node may be a single system or a cluster. A resource being managed can be a network, system, or an application. Being mostly used in academic environments, “best effort” was and is a sufficient policy for committing resources to users performing their computational workload. Moving into the commercial space, “best effort” is no longer sufficient [19], businesses will be bound by commitments.

4.1.3 Promise Theory

Promise theory is a set of assumptions and a language for describing promises made between autonomous agents [8].

Requirement 1 (Voluntary Cooperation) states: Agents (i.e. humans, computers or indeed any entity that can be reasonably thought of to hold some kind of promise, whether actually, vicariously or by association with its owner or designer) are said to be autonomous if they cannot be forced to make any promises about their behavior by any outside agent.

This assumption means that agents can only extend their influence on others by making personal promises that ensure others will value their promises and cooperate voluntarily. This ‘extreme' viewpoint is a somewhat realistic model of how people, businesses and even technology actually behave in the final analysis. Some readers will find it cynical, but cynicism is based on worst case realism, and so one can think of it as embodying the assumption of risk. More pragmatically, it forces us to document every condition for cooperative behavior between the parts of a system that could (for any reason) cease to behave in the manner we might prefer for the good of the business.

Promise theory agents are therefore impenetrable to outside influence, possess private knowledge, and the promises that they make to one another cannot be forced onto them by someone outside's will. This apparent limitation confuses newcomers into thinking that this is a flaw in the theory. After all, everyone knows that when the boss commands, slaves obey. However, this is not a flaw in promise theory but a strength: command is an illusion that survives only in systems where voluntary cooperation can be taken for granted. The challenge in promise theory is to show how or why such a command could be realistically given and obeyed, given the fact that no agent or intermediary in the chain of promises is always willing or able to keep all of its promises (even with the best of intentions).

Promises between agents can deal with things like QoS, quality of behavior, specifications of state, etc. For instance one has authorizations (promises to grant access) and obligations (promises to follow up on a different promise) or dependency, etc. These can all be translated into the notion of promises.

The key assumption of promise theory is that of autonomy. Agents (i.e. humans, computers or any entity that can be associated with a promise even if by association with its owner or designer) are said to be autonomous if they cannot be forced to make any promises about their behavior by an outside agent.

A promise usually represents voluntarily proposed behavior, represented as a directed relationship between two agents. Agents are any convenient set of separable components in system. Often the set of agents will be a maximal separation of concerns within a software system, but this need not be the case. It is up to the modeler to decide how the agents are best represented. This 'extreme' viewpoint is a realistic model of how people, businesses and even technology actually behave in the final analysis and a useful abstraction of limited determinism in control systems. The challenge in promise theory is to show how or why a desired service would be performed by a remote agent, given the fact that agents must necessarily be both willing and able to do so.

A promise with body +b is understood to be a specification to exhibit or “give” behavior from one agent to another (possibly in the manner of a service), while a promise with body -b is a specification of what behavior will be “received” or “used” by one agent from another (see Figure 23). The body of a promise is assumed to have a type which uniquely identifies that subject of the promise.

Symbol	Interpretation
$a \xrightarrow{+b} a'$	Promise from a to a' with body b
$a' \xrightarrow{-b} a$	Promise to accept b
$v_a(a \xrightarrow{b} a')$	The value of promise to a
$v_{a'}(a \xrightarrow{b} a')$	The value of promise to a'
$\oplus$	Combination of promises in parallel
$\otimes$	Combination of promises in series

Figure 23: Promise Notation

A conditional promise body is written X/Y if we promise X subject to the condition that Y is promised or delivered by another agent. Finally, a promise valuation as shown in Figure 24

$$\left(a_j \xrightarrow{b} a_k\right)$$

Figure 24: Promise Valuation

is a subjective interpretation by agent ai (in any local currency) of the promise in the parentheses. Usually an agent can only evaluate promises in which it is involved, as promises are only received (observable) by their recipient and their giver.

We could define Grid system as a collection of servers, potentially belonging to multiple administrative domains, collaborating as a distributed system and appearing as a single virtual machine to the end user. Agents in a Grid all run common software which binds them together in a standardized way. A single agent could be part of several Grids.

According to this definition a Grid has an end-user or a single point of dispatch for a unit of work (see Figure 25). In other words, each user exchanges promises with only a single agent that represents all others. This is referred to as a virtual domain. In a homogenous Grid any one of the agents can play this role. The remainder of the Grid promises to work therefore something like a load-scheduling structure. We do not have to assume complete symmetry between the servers, but we shall assume that any node in the Grid can play this role. Certain servers might be able to deliver services that others cannot, it is up to the internal agreements to decide how these resources will be dealt.

As depicted in Figure 25, a distributed system in which every node can be the initiator of a request to the entire group requires N(N-1) promise bindings to standardize a communication infrastructure, i.e. O(N2). An ad hoc service delegation to a single entry point requires typically only O(N) promises. The topology of the graph determines the complexity of cooperation.

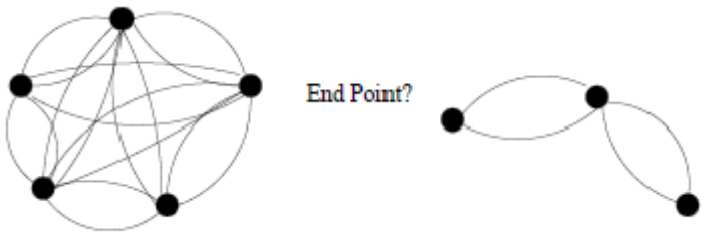


Figure 25: Single Agent Representing All Others

Let S be a server, C be a client and I be an intermediate agent. Virtualization is a translation service promised conditionally by I which reveals to the C an interface to base-service provider S, if and only if base-service is delivered to the intermediate agent.

In this virtualization scenario, 'translation' is desirable. The virtualization-layer agent's task is to translate the service from S to the client C. Delegation and virtualization of a task to another server requires a minimum of promises to send and accept the data ±d as mentioned in Figure 26. The initiating agent will then provide some kind of translation t(d), dependent on the data received t(d)/d, and confirm the usage of the dependency −d to the end user. In our model, virtualization is simply a form of delegation.

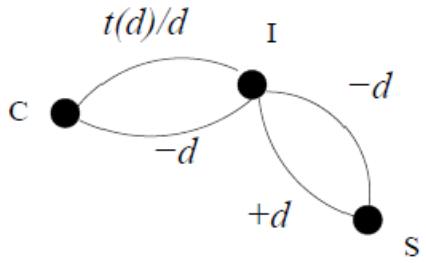


Figure 26: Virtualization

One of the difficult problems of policy consistency is in transferring responsibilities from one VO to another especially when a VO acts as an interloper for another. Consider VOs A, B and C, and suppose that B has some resources which it can promise to others.

The difficulty in this promise is that the promise itself refers to a third party, and this mixes link-worlds with constraints.

In a Grid environment users and resource providers, often belonging to multiple management domains are brought together. Users must be given some form of commitment and assurances on top of the allocated resources, such as performance, security, availability, latency, throughput, variance etc. (sometimes also referred to as QoS) as well as in terms of responsibilities for dealing with erroneous conditions, fail-over policies or backups and more.

Those terms need to be agreed upon before use and manifested in form of promises. Negotiating a promise is an exchange (protocol) of messages between user and provider, potentially involving some form of a middleman or broker. The result is an agreement of all terms that are important for either side before signing a contract or signing an obligation.

It is important to understand a typical hosting environment to understand which kind of promises between a Grid user and a resource provider has to be defined. These promises demand additional expertise in various level of escalation. A more common situation is when a user has an SLA with a provider and the outcome of that SLA can dependent on a third provider. How do the consumer and provider agree on the outcome of the SLA? Since the two are in different management domains, the provider may argue that its SLA has been met while the user may argue that it has been violated. This requires agreement protocols that will repudiate and ensure that the two parties agree on the outcome according to technical components (service description, metrics, SLA/QoS parameter, etc), organizational components (*e.g.*, level of escalation, maintenance, reporting, change management) and legal components (*e.g.,* modes of payment, propriety rights, legal responsibilities).

4.2 Final System Design

Since distributed systems are coined as Grids, in particular, these have been widely discussed in recent times. Today the new term "Cloud Computing" is emerging but the principles covered are the same as those below. This defines a Grid as:

"A collection of servers, potentially belonging to multiple administrative domains, collaborating as a distributed system and appearing as a single virtual machine to the end user. Agents in a Grid all run a common software which binds them together in a standardized way. A single agent could be part of several Grids."

A Grid service is any service that can be provided on the Grid. Each service described in the Open Grid Services Architecture (OGSA) [14] is a single Grid service or a composition of Grid services. In fact, these services implement the functionality which is provisioned to the customer. They can also offer management functionality for other services on the provider side [5].

A Grid has an end-user, client or a single point of dispatch for each task. In other words, each user exchanges promises to perform work ultimately with a single agent that represents all others. In a homogenous Grid any one of the agents can play this role — it is not fixed, as it might be in a traditional hierarchical system. The remainder of the Grid promises to work therefore like a load-scheduling structure on behalf of the client. We do not have to assume complete symmetry between the servers, but we shall assume that any node in the Grid can play this role. Certain servers might be able to deliver services that others cannot, it is up to the internal promises or agreements to decide how these resources will be dealt.

The identification of the single point of dispatch does not imply a hierarchy of control and coordination, only a recognizable interface for each potential client in the Grid, but the mesh promise-graph symmetry tells us that every node in a Grid must share the same singular coordination mechanism (implemented usually by the "middleware") to participate in the collaboration. There is thus one rigid kind of promise channel that must exist between all agents, in a complete graph, to coordinate jobs.

Promise theory indicates that the number of promises required to maintain a Grid infrastructure is relatively expensive compared to the necessary number required to get any particular job done. Every promise made by an agent is something that requires some work to deliver on or to verify so making such a mesh is a daunting investment (which perhaps deters businesses). But once the mesh of promises is in place it opens the door to great flexibility if the agents can manage the cost as shown in Figure 27. Perhaps this might be the driver for business adoption.

Grid architecture can be viewed as a high-level design consisting of virtual organizations (VOs) and their interconnections. In Grid environments the concept of virtualization is very important to express the flexibility of exchanging services and resources between agents and Grid-levels. VOs can be defined as a set of virtual nodes (resources) and virtual services that can be used by users (individuals) and/or customers (a group of users or organizations) as members of a VO, to achieve a common goal. Furthermore, resources and services of a VO may also be provided to members of other VOs as presented in Figure 27 to achieve optimal costs of a delegated service(s) or sub-service(s).

Dealing with complex virtual services and virtual resources where service compositions have to take place on-demand, for certain periods of time, and across organizational boundaries imposes new challenging requirements to businesses especially because the underlying accounting systems are often ill-equipped to support this [11]. Grid systems have evolved over time from pure computation (with little instrumentation and which benefit from a maximal mesh) into service Grids which have simpler needs but provide a sustainable platform for electronic service provisioning in research-oriented and commercial multi-domain environments.

Thus our first question to be answered is why would anyone need to link together geographically disparate computing systems into a tree-like hierarchy, with a single point of dispatch and return when they could possibly do everything from a single location?

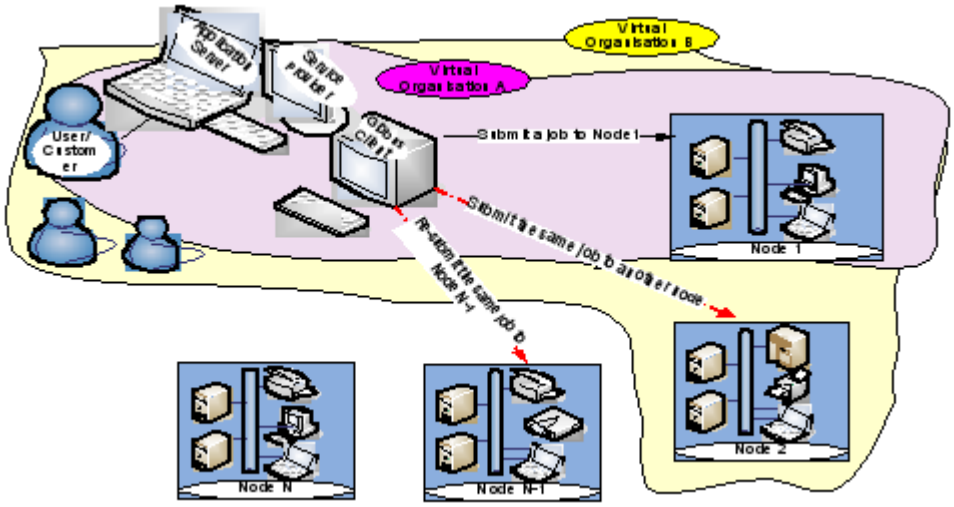


Figure 27: Business Promises

4.2.1 Centralization and Hierarchy

In [9] one of us argued that the main reason for centralization in distributed computing is the calibration of distributed results, i.e. not the coordination, but the ultimate comparison and assembly of results from other nodes around the system in a consistent fashion. If only coordination is needed, then any kind of mesh will do. Without the need to bring together and compare computations from many sources under a common set of standards, there is no technical or economic need for having a “collating” or management front-end to distributed services. In order to provide a service, however, there has to be a point of service, which necessitates this kind of coordination. So why not simply use a traditional centralized, fixed infrastructure approach? When we are looking for business value from a Grid architecture, a justification must include the need for this process of calibration, i.e. making sense of the service-results arriving from disparate sources in a common framework.

In a traditional distributed system based on fixed relationships, the cost associated with centralization is low for the satellite or “client” nodes but high for the central hub, so there must be sufficient added value from the centralization to make the cost economically valid. In traditional HPC, massive parallelization was a sufficient reason for this kind of approach: a single agent in the network had a task requiring great resources which could not be completed by any single agent in reasonable time. Thus the economic advantage was the speed of the result thanks for parallelization — and it made economic sense to equip the central node with the resources to process input from all the clients quickly and coordinate. After all, some services are only viable if they can be provided on time (*e.g.*, forecasting). Thus the cost of calibration is offset by the saving on delivery time in any centralized system. The only additional functionality that Grid adds is the ability to change the configuration of relationships. So the question becomes: is this something businesses need to do sufficiently often?

Grids are often hierarchical in design. Promise theory suggests that there is a simple economic reason for the emergence of hierarchical organization [3]. Hierarchies begin with the need to communicate between multiple parties to complete a task (like in Grid, but also

any other distributed system). In a simple “request-reply” end-to-end service relationship, there is no need for coordination or calibration of data, the results are always consistent as they always come from a single source.

Centralization with a single master node however allows results from the parallel work of many slaves to be coordinated at the minimum cost to the slaves. The price is that of a higher cost for the central calibrating node. This organization allows the slave nodes to focus on their specialized task without wasting resources on intercommunication. The alternative construction would be a mesh of peer to peer connections required for full connectivity and would be more expensive for all nodes, requiring N(N-1) promises rather than the 2N required for a central organization. Ultimately however, the cost of coordinating by routing messages and calibrating data to a common standard becomes too great for a single node to bear, and that coordination work also must be divided amongst a layer of “middle management”, which in turn has a single coordinating node. Thus a new layer of the hierarchy emerges through necessity. This purely economic process leads to a tree structure that is a trade-off between depth and width costs (see Section 4.3).

4.2.2 Service Orientation

Current programming and economic models refer to the Service Oriented Architecture (SOA) in the same sentence as Grid as if these ideas were similar. We feel that this is somewhat misleading as all distributed systems are essentially similar and the value of calling them by different names likes in distinguishing them else they are simply the same. Hence we would like to make a clearer distinction between Grid and SOA.

The SOA is defined here as follows:

“A collection of distributed services belonging potentially to multiple administrative domains, which interact peer-to-peer, using well-known protocols and standards. Nodes in SOA do not have to run common software, but share compatible services opportunistically.”

SOA is much less rigid than a Grid. If a Grid is like a crystal, then SOA is an amorphous plastic of tenuous links [10], [3]. The agents in an SOA do not have to be tightly coordinated by middleware in the same way as Grid because they use open standards and interfaces. The protocols mentioned for SOA are commonly based on Web Services, but this is not a requirement. In SOA, coordination is ad hoc: there is no local or central dispatcher or point of coordination for a job. Nor is there common software which maintains Grid-wide state information. Whichever agent requires a service from another is free to ask on a peer to peer basis. In the first analysis, this seems to make SOA much simpler than Grid, since one only needs of order one promise from each service point to each client that wants to use it, in the manner of a random graph. This sparse view could be misleading however. The problem with SOA is service discovery and the fact that it is principally one-to-one. Each node needs to know which services exist. For a businesses deploying a service organization, service discovery is part of regular business process and requires no special technology: the marketplace is the place of discovery, or the service is ordered from a directory. Grid offers a way of solving service coordination automatically and so it is capable of greater flexibility and retooling when new and different tasks come along. The question remaining is whether such retooling is justified.

To understand why businesses might forego a cheaper approach to service management like SOA or traditional in-house hierarchy for something apparently more expensive like a Grid, we have to look at the incentives that balance the cost of infrastructure with cost of coordination: the need for rapid adaptation to change with reliability.

4.3 Implementation

To be able to keep or deliver on a Service Level Promise, there must be an underlying technology capable of performing appropriately. The more individual components that are involved in such a system, the less must be certain its abilities to deliver must be. Independent uncertainties accumulate according to Pythagoras' theorem. Our methodology allows us to perform rigorous accounting of the uncertainties associated with each separate promise.

An architecture proposed by us for Grid Management is a layered understanding of the Grid hierarchy (see Figure 28) showing the service layers that contribute the different elements of management. This architecture assumes that the Grid is sufficiently large to warrant the overhead of all the layers. For smaller projects promise theory indicates that simple service orientation is a superior approach. The trouble with the layered architecture approach is that layering patterns are designed to hide detail, and this also hides the uncertainties making the system harder to analyze.

The service model is in a sense a way to avoid hierarchical structure, so one can ask the question: is the service level management approach at odds with the typical layered approach to designing computing management systems? The answer is both yes and no: it is one of scale. What promise theory allows us to do is strip away the assumptions of management models and deal only with the individual capabilities and needs of the parts of the system. We do this as follows.

Independently changeable components in a system are defined (degrees of freedom) and representing them as autonomous agents. We assume that these agents cannot be force into compliance by any outside entity, nor can one agent "demand anything from another" or impose obligations. This means that the only kind of behavior left is voluntary behavior and the challenge remaining is to document a sufficient number of promises to achieve the desired system functionality at the appropriate level of service (SLA or Service Promise). Then we systematically document every promise that is needed to guarantee the desired behavior (assuming the promises are kept). Each promise has a cost and a value associated with it which can now be counted. This generates naturally a selfish agent game-theoretic approach to the economics of the system. We use these valuations to analyze the economics of the system [8].

The local economics of network relationships are now quite simple and depend mainly of the topology at a point. We need to show how the cost of a particular topology impinges on the cost of either coordination or calibration. We should recall that promises are not about continuous network communications, so the cost of making a promise is entirely in the establishment of the promises. The maintenance of the promise depends on its type however. Promises that requires an exchange of information between agents involve propagation of data, which introduces measures of time-taken, latency etc. Promise theory thus predicts a two-dimensional model for economic value.

Promise graphs as mentioned in Section 4 form networks and the economics of coordination thus have two facets: cost and efficiency. The cost of establishing promises increases with the number of promises since each promise generally requires some behavior or work to be done by the agent. The efficiency of coordination involves communication and therefore has to do with propagation of effect over the coordination distance (this is a network depth issue). We can divide the discussion into what is good for the group and what is good for the individual agent.

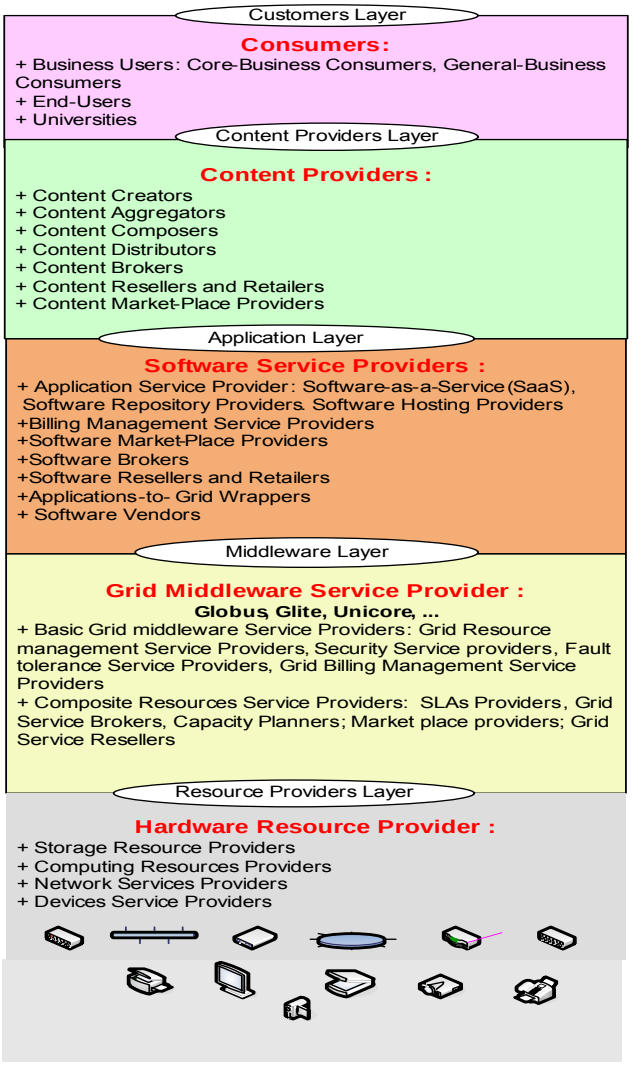


Figure 28: Grid Layers and Roles of Grid Participant (Agent)

There are two extreme cases for topological connectivity in a global region and a range of values in between. These are the complete graph (all pairs nodes linked peer to peer with N(N-1) directed links) and that of centralized hub (with (N-1) nodes linked directly to a

single hub, making 2(N-1) directed links). If we assume that, to a first approximation, agents are homogeneous and value promises from one another equally then the cost and value of promises is proportional to the number of promises.

If agents do not use route messages for each other (requiring many coordination of promises), they have to coordinate with every other agent individually in a complete graph of N(N-1) promises of each type of promise in the ensemble of size N. If they network their efforts however into a hub or chain then they can reduce their promises to order 2(N-1) in total, but now there is a new issue: depth or efficiency. In a Grid, of course, some routing is likely, especially with multi-layered architecture. Middleware is by definition a message routine architecture.

Depth versus width is an economic trade-off. Greater centralization reduces depth and hence increases the coordination efficiency, but it increases the cost burden of promises at the hub. The costs are in-homogeneously distributed. In a chain (the opposite of a hub), the cost of keeping promises is maximally distributed but the depth is maximal too, meaning low coordination efficiency and delays.

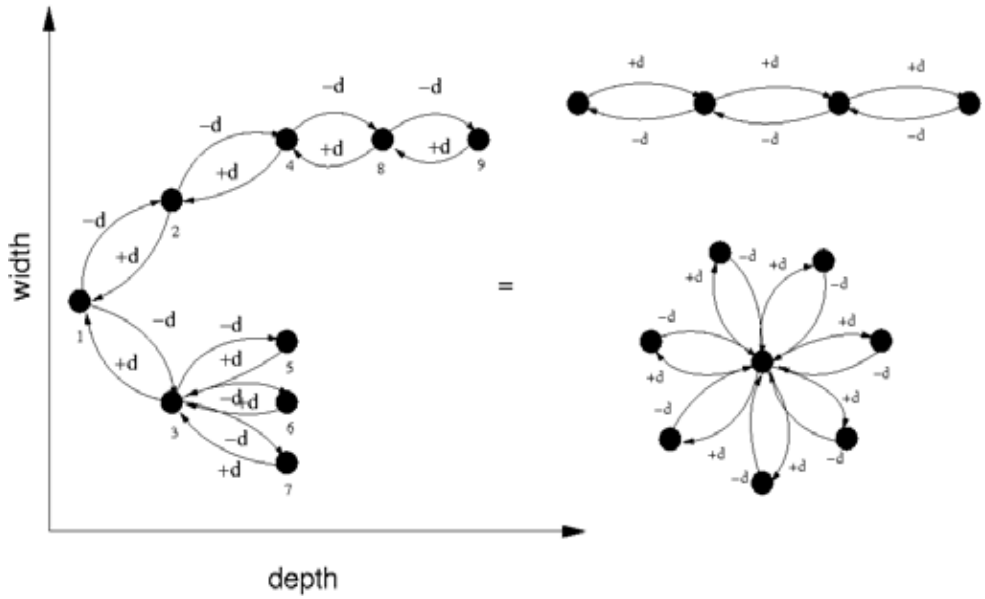


Figure 29: Bilateral Communication Structures Indicating Depth and Width of Promise Bindings (Tree is Between Extremes of a Chain and a Hub) [11]

4.4 Evaluations

Up to now, the enthusiasm for Grid has been weaker in business than in academia, most likely because of its overhead, heterogeneity and a lack of trust in the Grid technology. A further reason might be that people simply do not understand what Grid actually offers above and beyond the alternatives. We suggest that reasons for this might include the uncertainty and apparent cost of making promises to many different agents, not all of whom are necessarily reliable, but also the protection of intellectual properties when sharing resources between different sites. Are Grids private or formed from consortia of companies?

The number of promises that needs to be kept and verified is very large ($O(N^2)$) in a traditional understanding of what is meant by Grid. The only possible cost justification for business to adopt it would be a scenario in which hardware and software configuration were constantly being redefined and an automated redeployment were necessitated. Today's commodity computing is built more around cluster technology than Grid, and the increasing mention of SOA indicates that the original Grid model is not sufficiently well-defined, but it is true that the distinctions are still perhaps deliberately vague in the literature, which hinders true progress in this area.

The SLA has become the dominant tool for discussing performance of IT and management systems in businesses and computer science — but opinion indicates that it is not a very good tool because it offers no methodology. Promise theory on the other hand offers a methodology for analyzing systems in a way that is compatible with service orientation.

4.5 Modeling and Implementation Achievements and Key Results

This work contributes to a systematic categorization of promises with a particular analysis of existing SLAs and helps to identify responsibilities for critical IT processes during the execution of SLAs in large scale distributed systems such as Grid computing. The middleware approach to design of IT systems leads to a hierarchical structure that has economic implications for scalability. We can characterize the scaling properties of

distributed systems in a two-dimensional scale including a trade-off between depth and width. This is an economic game and its result depends on the size of the system. Layered architectures are expensive and will only pay off on a large scale. For smaller systems, more ad hoc network structures are cheaper — this is often referred to as the Service Oriented Architecture.

The advantage of the promise approach to modeling is that is does not build in any initial assumptions about efficiency, scaling or cost so one will arrive at an impartial judgment about the architecture that best accommodates the conditions under which the system operates.

Furthermore, we have to distinguish between promises and the usual idea of obligations and unequivocally notice that promises are a simpler theoretical notion and a more practical tool than obligations in the reduction of an agent's uncertainty about the behavior of other agents.

A Grid has an end-user, client or a single point of dispatch for each task. In other words, each user exchanges promises to perform work ultimately with a single agent that represents all others. In a homogenous Grid any one of the agents can play this role - it is not fixed, as it might be in a traditional hierarchical system. The remainder of the Grid promises to work therefore something like a load-scheduling structure on behalf of the client. We do not have to assume complete symmetry between the servers, but we shall assume that any node in the Grid can play this role. Certain servers might be able to deliver services that others cannot, it is up to the internal agreements to decide how these resources will be dealt.

Note: Based on organizational and administrative problems within the affiliation of Oslo University College (HIO), the content of this section no. 4 does not include final information, thus, it describes the current incomplete state of the SaPDoGS project only. This was mainly caused by personnel problems and employment issues, which, however, does not effect all other WP8-internal projects in 2008 and 2009. Furthermore, next steps in terms of the project will not be undertaken within WP8 any more, including no claims for any funding.

5 BP3EM: Best Practices, Processes and Promises in Economic Management

In IT Management, more and more attention is paid to best practice recommendations and process-oriented approaches like the IT Infrastructure Library (ITIL) - as already described in Deliverable D8.2 [30]. This turn from a pure technological view point into IT Service Management (ITSM) from a perspective covering organizational aspects comes with various challenges. The overall goal of BP3EM is to contribute in bridging the gap between organizational ITSM approaches and the “hands-on-the-keyboard” system management and administration tasks.

5.1 Final System Design

The output of BP3EM is a comprehensive document reflecting the results achieved in more detail than covered by this project report. This document, entitled *Integrating cfengine, ITIL and Enterprise Processes* [12] shall not be regarded as a final solution, but contains a collection of ideas and concepts on integrating ITSM and system administration.

5.1.1 Scope

The goals of ITIL and the purpose of cfengine are quite different: ITIL gives recommendatory guidance in process- and service- oriented IT Service Management, while cfengine provides a powerful solution framework for a variety of common network and systems administration tasks. In other words:

- The scope of ITIL is much broader than traditional systems administration, but: Portions of systems administration and configuration management tasks take place in the context of certain ITIL processes.
- Cfengine was not designed to replace ITSM tools like trouble ticket systems (TTS), workflow management or CMDBs, but: in the more technical areas of IT Service Management, cfengine is able to support ITIL processes in their activities. One of the goals BP3EM is to give an overview on how cfengine can be used to support selected IT Service Management tasks according to ITIL.

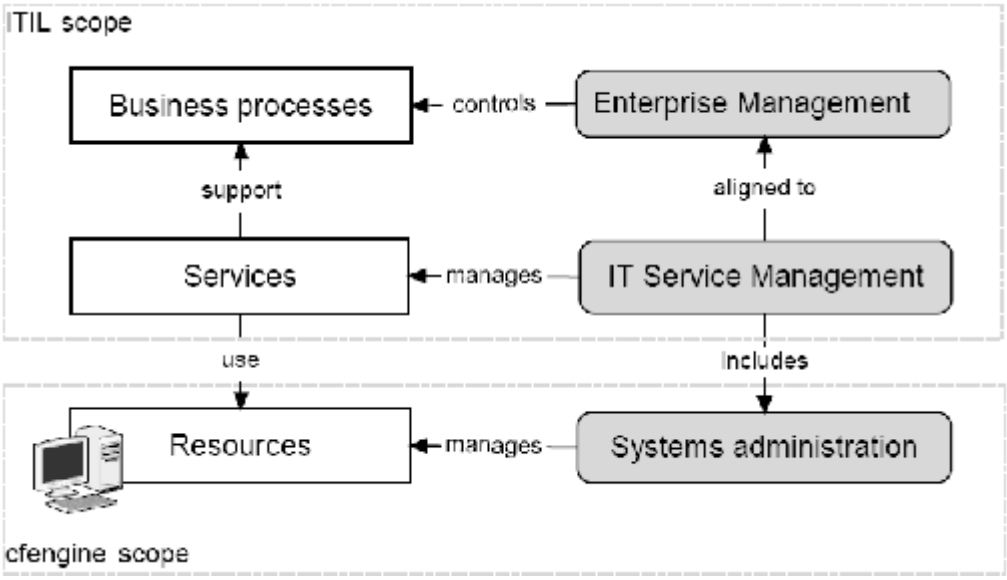


Figure 30: Scope of ITIL vs. Scope of cfenging

Figure 30 illustrates the different scope of ITIL and cfengine in a comparative manner. In particular, it shows where and how IT services and resources and respectively IT Service Management and system administration correlate and thus provide a potential for vertical integration.

5.1.2 ITIL Objectives and Concepts

The IT Infrastructure Library (ITIL) [20] is a collection of books, in which best practices for IT Service Management are described. Today, ITIL can be seen as a de-facto standard in the discipline of ITSM, for which it provides guidelines by its current core titles Service Strategy [24], Service Design [22], Service Transition [25], Service Operation [23], and Continual Service Improvement [21]. ITIL follows the principle of process-oriented management of IT services.

In effect, the responsibilities for specific IT management decisions can be shared between different organizational units as the management processes span the entire IT organization independent from its organizational partition. Whether this means a centralization or decentralization of IT management in the end, depends on the concrete instances of ITIL processes in the respective scenario.

ITIL has its roots in the early 1990s, and since then was subject to numerous improvements and enhancements. Today, the most popular release of ITIL is given by the books of ITIL version 2 (often referred to as ITILv2), while the British OGC (Office of Government Commerce), owner and publisher of ITIL, is currently promoting ITIL version 3 (ITILv3) under the device “ITIL Reloaded”. It is important to understand that ITILv3 is not just an improved version of the ITILv2 books, but rather comes with a completely renewed structure, new sets of processes and a different scope with respect to the issue of IT strategies, business IT alignment and continual improvement.

- Key Concept 1 (Service): A service is a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks. ITIL differentiates between utility (ready for purpose) and warranty (ready for use) of an (IT) service.
- Key Concept 2 (Management process): The set of management processes of ITILv3 includes (excerpt, in alphabetical order):
  → Access Management
  → Availability Management
  → Capacity Management
  → Change Management
  → Evaluation
  → Event Management
  → Incident Management
  → Information Security Management
  → IT Service Continuity Management
  → Knowledge Management
  → Problem Management
  → Release and Deployment Management
  → Request Fulfilment
  → Service Asset and Configuration Management
  → Service Catalogue Management
  → Service Level Management
  → Service Validation and Testing
  → Supplier Management
  → Transition Planning and Support
- Key Concept 3 (PDCA): ITIL uses the Deming quality circle as a model for continual quality improvement, where quality both relates to the provided IT services (cf. key concept 1) as well as the management processes deployed to manage these services (cf. key concept 2). Continual improvement as to ITIL means to follow the method of Plan-Do-Check-Act (PDCA):
  → Plan: Plan the provision of high-quality IT services, set up the required management processes for the delivery and support of these services, define measurable goals and the course of action in order to fulfill them.
  → Do: Put the plans into action.
  → Check: Measure all relevant performance indicators, and quantify the achieved quality compared to the quality objectives. Check for potentials of improvement.
  → Act: In response to the measured quality, start activities for future improvements. This step leads into the Plan phase again.

5.1.3 cfengine Objectives and Concepts

Cfengine is a free software package for automating the installation and maintenance of networked computers. The project began in 1993 and it has been in widespread use since 1995. Cfengine is available for all major Unix and Unix-like operating systems, and it will also run under NT-derived Windows operating systems via the Cygwin Unix-compatibility environment/libraries.

Cfengine scales easily from a single host to tens of thousands of hosts. As of this writing, the largest installations we know of regulate around 20,000 machines under a common administration. Cfengine can manage many aspects of system configuration and maintenance, including the following:

- Performing post-installation tasks such as configuring the network interface.
- Editing system configuration files and other files.
- Creating symbolic links.
- Checking and correcting file permissions and ownership.
- Deleting unwanted files.
- Compressing selected files.
- Distributing files within a network.
- Automatically mount NFS file systems.
- Verifying the presence and integrity of important files and file systems.
- Executing commands and scripts.
- Applying security-related patches and similar system corrections.
- Managing system server processes.

Cfengine's purpose is to implement policy-based configuration management. In practical terms, this means that cfengine greatly simplifies the tasks of system configuration and maintenance. For example, to customize a particular system, it is no longer necessary to write a program which performs each required action in a procedural language. Instead, you write a much simpler policy description that documents how you want your hosts to be configured. The cfengine software determines what needs to be done in terms of implementation and/or remediation from this specification. Such policy descriptions are also used to ensure that the system remains configured as the system administrator wishes over time.

As stated, cfengine operates on hosts in order to bring their configurations in line with the specified policies. The following terms represent the key concepts building the foundation for cfengine:

- Key Concept 1 (Host): A host is a single computer that runs an operating system like Unix, Linux or Windows. We will sometimes talk about machines too, and a host can also be a virtual machine supported by an environment VMWare or Xen/Linux.
- Key Concept 2 (Policy): This is a specification of what we want a host to be like, or how we want it to behave. A policy is essentially a piece of documentation that describes technical details and characteristics. Cfengine implements policies that are specified via directives of the sort we just considered.
- Key Concept 3 (Configuration): The configuration of a host is the actual state of its resources, *e.g.,* the permissions and contents of files, the inventory of software installed, etc. It is the 'state of affairs' on a particular host at a given time.

5.1.4 Using cfengine to Implement ITIL Objectives

Cfengine users are interested in the ability to manage, i.e. cope with system configuration in a way that enables a business or other organization to do its work effectively. They do not want reams of human management because this is what cfengine is supposed to remove. To be able to use ITIL to help in this task, we have to first think of the process of setting up as a number of services. What services are these? We have to think a little sideways to see the relationship.

- Service: Providing a sensible configuration policy, responding to discovered problems or the needs of end-users.
- Change: An edit of the configuration policy, with appropriate quality controls.
- Release: A new configuration policy, consisting of many changes. A new version of cfengine? This could be a major and disruptive change so it should be planned carefully.
- Capacity: Having enough resources for cfservd to answer all queries in a network. Having enough people to support the processes of deploying and following cfengine's progress. You should keep this kind of thinking in mind, and train yourself to see every part of a task in “ITIL clothes”.

The final BP3EM [12] framework consists of five main parts:

1 An introductory overview of cfengine and its development (see Section 5.1.3 for a short abstract/summary)
2. An introductory overview of ITIL and its development (see Section 5.1.2 for a short abstract/summary)
3. A meeting of mind sets as a starting point for a more focused examination (see Section 5.1.4 for a short abstract/summary)
4. Guidance on using cfengine to implement ITIL objectives (see Section 5.2 for a more detailed overview)
5. A mapping of terminologies (see Section 5.3 for a more detailed overview)

5.2 Implementation

5.2.1 A Road-map for Adoption

BP3EM provides a checklist of ITIL compliant steps for using cfengine in a machine life-cycle:

1 Set up cfagent running at scheduled interval in accordance to SLAs.
2. Set up versioning of policy.
3. Set up delegation of authorship.
4. Run cfenvd for passive monitoring.
5. Run cfagent for active monitoring.

The steps required to release cfengine into the operating environment under consideration of ITIL's Configuration Management, Change Management and IT Service Continuity Management are as follows:

1 Select installation medium, *e.g.,* DVD, net-boot with hooks to cfengine.
2. Start with essential promises, and formulate the configuration policy.
3. Use ITIL recommendations for deciding and refining configuration promises.
4. Evaluation and monitoring of promises using cfagent and cfenvd.

5. Use cfagent for monitor changes using cryptographic checksums.
6. Develop recovery plans as referred to by ITIL Continuity Management.
7. Use cfengine to automate backup of data and automate the duplication of servers for load balancing and redundancy.

5.2.2 Exemplary Implementation Analysis: Incident and Event Management

Cfengine employs the idea of continual maintenance. ITIL, on the other hand, moves from release to release and does not recognize the effect of gradual entropic decay of state. Instead ITIL deals with incidents which must be corrected. While it is true that these incidents are maintenance, the repairs are more costly to initiate if they occur as exceptional events than if they are considered on a regular basis.

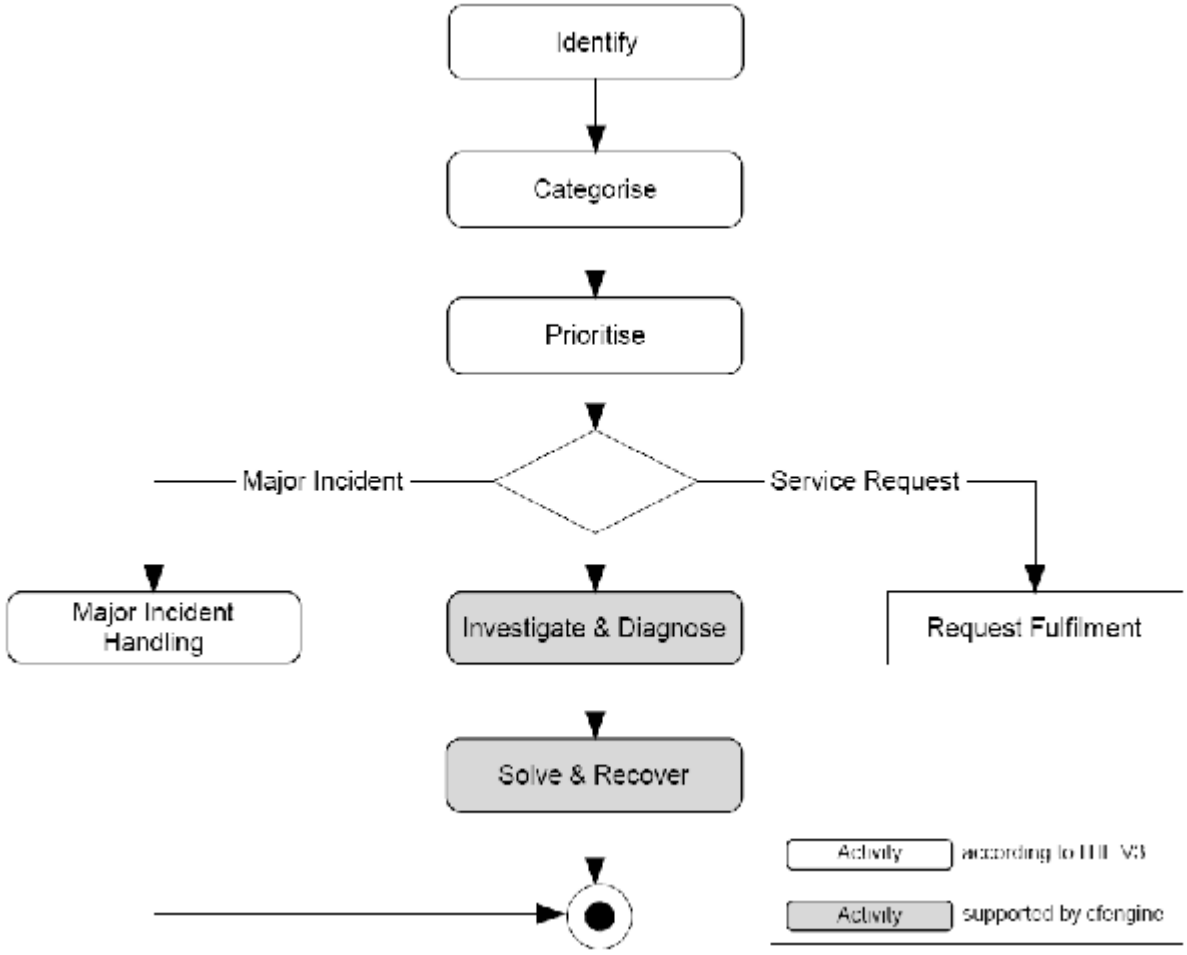


Figure 31: Exemplary Incident Management Process in Compliance with ITIL V3 Using cfengine Support

Figure 31 and Figure 32 show ITIL processes for the handling of incidents and events. They show the aspects of dealing with events that are mainly human oriented (i.e. need major human intervention/task processing), and those events in shaded boxes that can be automated using cfengine.

In both figures we see that there must be a basic monitor at the top of the process chain which is responsible for observing events. This fits well with the view of promise theory in which a neutral observer is required to measure the state of different component agents in the system. Not all events are necessarily relevant or interesting; so we can filter these

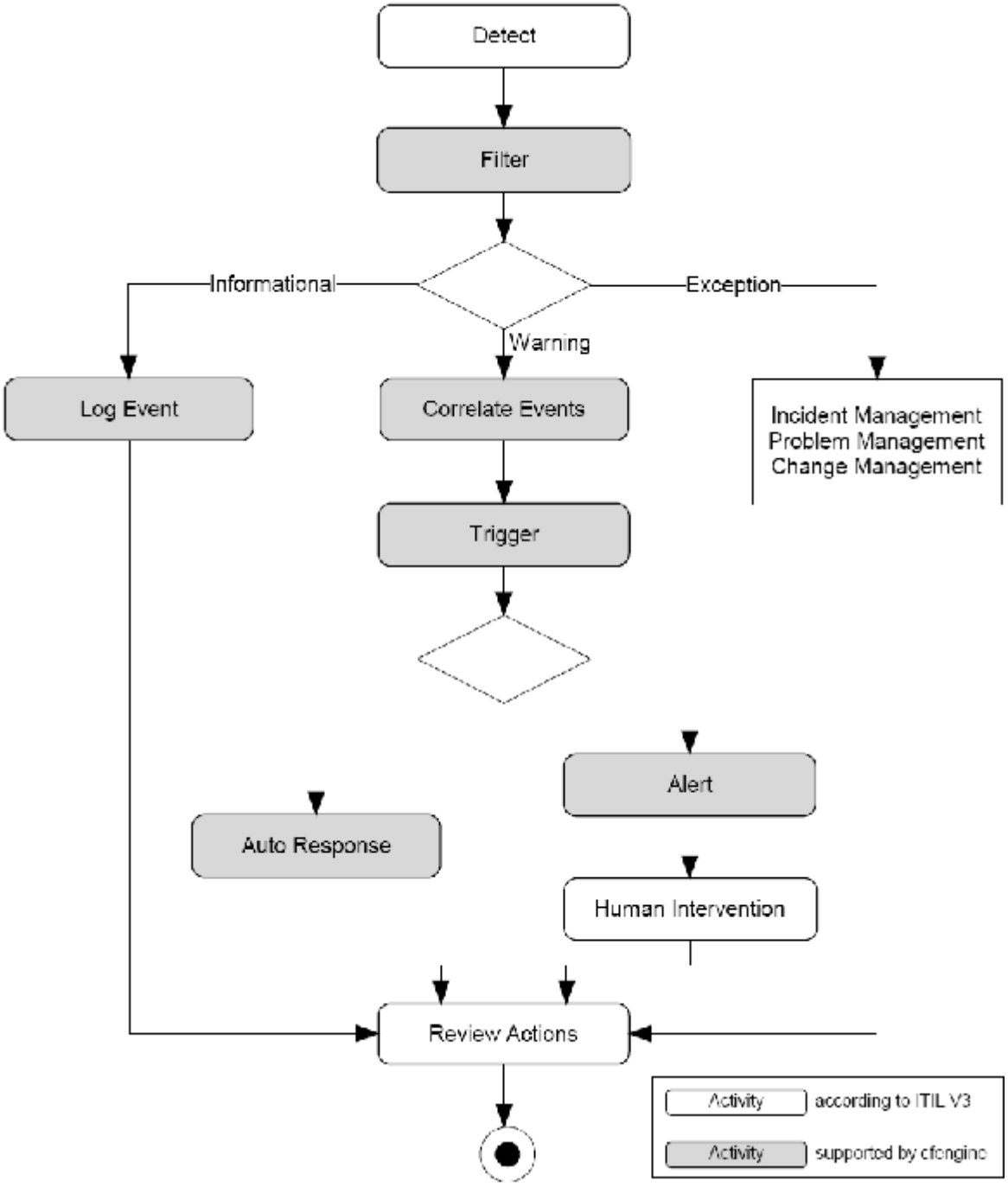


Figure 32: Exemplary Event Management Process in Compliance with ITIL V3 Using cfengine Support

based on a policy. Cfengine's event monitors come from two sources: cfagent (for monitoring the state of promises which are being managed) and cfenvd (for passively monitoring the environment).

5.3 Evaluations

In order to assess and evaluate the potential degree of integration between ITIL and cfengine more carefully and in a more comprehensive way, BP3EM does not only focus on processes (in the sense of workflows), but also includes a mapping of terminologies. This section shows an excerpt of this mapping.

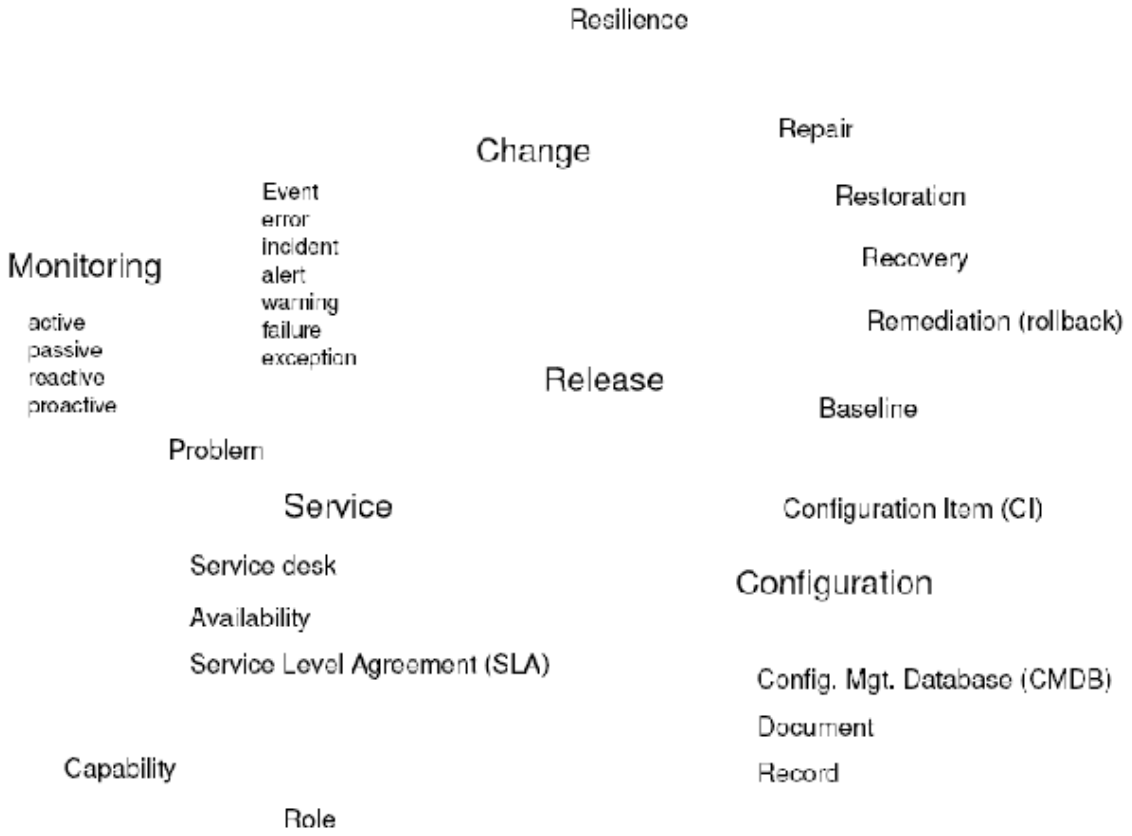


Figure 33: ITIL Terminology

Figure 33 illustrates some of the key terms defined and used by ITIL. It was the goal of the evaluation phase of the BP3EM project to find correlations between these terms and a system administrator's point of view (assuming the system administrator uses cfengine). Thus, BP3EM provides a terminology mapping in order to pave the way for an integrated approach. An excerpt of this mapping is presented in the following:

- Active Monitoring:
  → Monitoring of a configuration item or IT service that uses automated regular checks to discover the current status.
  → Cfengine mapping: Cfengine performs programmed checks of all of its promises each time cfagent is started. Cfagent is, in a sense, an active monitor for a set of promises that are described in its configuration file.
- Alert:
  → A warning that a threshold has been reached, something has changed or a failure has occurred.
  → Cfengine mapping: A cfengine alert fits this description quite well. Most alerts are user-defined, but a few are side effects of certain configuration rules.
- Baseline:
  → A snapshot of the state of a service or an individual configuration item at a point in time.
  → Cfengine mapping: In cfengine parlance, we refer to this as an initial state or configuration. In principle a cfengine initial state does not have to be a known-base line, since the changes we make will not generally be relative to an existing configuration. Cfengine encourages users to define the final state (regardless of initial state).

- Configuration:
  → A group of configuration items (CI) that work together to deliver an IT service.
  → Cfengine mapping: A configuration is the current state of resources on a system. This is, in principle, different from the state we would like to achieve, or what has been promised.
- Configuration Item:
  → A component of an infrastructure which is or will be under the control of Configuration Management.
  → Cfengine mapping: A configuration item is any object making a promise in cfengine. We often speak of the promise object, or “promiser”.
- Error:
  → A design flaw or malfunction that causes a failure.
  → Cfengine mapping: Cfengine often uses the term configuration error to mean a deviation of a configuration from its promised state. The ITIL meaning of the term would translate into “bug in the cfengine software” or “bug in the promised configuration”.
- Event:
  → A change of state that has significance for the management of a configuration item or IT service.
  → Cfengine mapping: The same basic definition applies to cfengine also, but cfengine makes all such events into classes, since its approach to observing the environment is to measure and then classify it into approximate expected states. Cfengine class attributes (usually from cfenvd) may be considered as event notifications as they change.
- Exception:
  → An event that is generated when a service or device is currently operating abnormally.
  → Cfengine mapping: A state in which configuration policy is violated (could lead to a warning or an automated correction).
- Failure:
  → Loss of ability to operate to specification or to deliver the required output.
  → Cfengine mapping: ITIL's idea of a failure is something that prevents a promise from being kept. Cfengine's autonomy model means that it is unlikely for such a failure to occur, since promises are only allowed to be made about resources for which we have all privileges. Occasionally, environmental issues might interfere and lead to failure.
- Incident:
  → Any event that is not expected in normal operations and which might cause a degradation of service quality.
  → Cfengine mapping: Cfengine's philosophy of convergence gives us only one option for interpreting this term, namely as a temporary deviation from promised behavior. A deviation must be temporary if cfengine is operating continually, since it will repair any problem on its next invocation round. Events which do not impact promises made by cfengine are of no interest to cfengine, since autonomy means it cannot be responsible for anything beyond its own promises.
- Passive Monitoring:

→ Monitoring of a configuration item or IT service that relies on an alert or notification to discover the current status.
→ Cfengine mapping: Cfenvd is cfengine's passive monitoring component. It observes system related behavior and learns about it. It assumes that there is likely to be a weekly periodicity in the data in order to best handle its statistical inference.

- Policy:
  → Formally documented management expectations and intentions. Policies are used to direct decisions, and to ensure consistent and appropriate development and implementation of processes, standards, roles, activities, IT infrastructures, etc.
  → Cfengine mapping: Cfengine's configuration policy is an automated set of promises about the static and runtime state of a computer. Roles are identified by the kinds of behavior exhibited by resources in a network. We say that a number of resources (hosts or smaller configuration objects) play a specific promised role if they make identical promises. Any resource can play a number of roles. Decisions in cfengine are made entirely on the basis of the result of monitoring a host environment.
- Warning:
  → An event that is generated when a service or device is approaching its threshold.
  → Cfengine mapping: A message generated in place of a correction to system state when a deviation from policy is detected. Note that cfengine is not based on fixed thresholds. All “thresholds” for action or warning are defined as a matter of policy.

The above mappings represent an extract of the entire terminology mapping of the BP3EM framework which covers in addition the following terms and concepts: Availability, Audit, Benchmark, Capability, Change record, Chronological Analysis, Configuration Management Database (CMDB), Document, Emergency Change, Monitoring, Proactive Monitoring, Problem, Promise, Reactive Monitoring, Record, Recovery, Remediation, Repair, Release, Request for Change, Resilience, Restoration, Role, Service Desk, Service Level Agreement, Service Management.

5.4 Modeling Achievements and Key Results

The final BP3EM framework shows, based on the examples of ITIL and cfengine, how IT Service Management and system administration can be aligned and – at least rudimentary – integrated. The effects on economics and user behaviors can be characterized as follows:

- Effects of BP3EM on economics: Integrating ITSM and system administration is an important goal of Business-Driven IT Management (BDIM). Service Level Agreements (SLAs) directly interfere with ITSM activities, and it is vital for an effective alignment between enterprise goals and IT operations that ways and means are available which enable a balanced and complementary approach for IT management through all its aspects including, but not limited to network and system administration tasks.
- Effects of BP3EM on user behavior: The BP3EM framework can be used by practitioners in the areas of ITSM (*e.g.,* ITIL Service Managers) and network/system administration (*e.g.,* cfengine users) as well as by CIOs who want to bridge the gap between these two worlds. But more important, the framework provides a starting point and conceptual guidance for examining other ITSM frameworks (*e.g.,* COBIT, eTOM) and other system administration tools/solutions with respect to the goal of integration. This aspect is valid for both practitioners and researchers.

6 PRIPOL: Pricing by Policies

In the network management area, the research community has directed some efforts developing mechanisms to deliver end-to-end QoS in the Internet. Mechanisms for network congestion prevention and solving, control of service subscriptions and invocations, and dynamic traffic engineering functions have been the centre of study in potential intra-domain and inter-domain network management solutions. Nevertheless, although these solutions have been proved to be efficient to guarantee QoS delivery, the requirements, implications and the incremental efforts to elevate their business value have remained almost unexplored. The ability to carry out business- and QoS-oriented network management introduces several challenging problems addressed in this project.

Initially, business strategies must be properly modelled with appropriate business indicators, pivotal for the management of policies. Secondly, business indicators should be monitored and modelled as functions of measurable parameters of the managed systems. Thirdly, the dynamicity of events occurring in the managed network should be constantly evaluated as to define proactive and corrective management actions enforced through policy. The creation, deployment and modification of policies in runtime should be devoted to optimize business value, all in all under QoS delivery constraints. These problems make this research highly challenging, mainly when we consider a holistic approach to optimize business value under different patterns of resources utilization, patterns of traffic exchange between administrative domains and diverse network topologies.

6.1 Final System Design

The concept inspiring the design of PRIPOL is that creation, deployment and modification of policies in runtime should be devoted to optimize business value, all in all under QoS delivery constraints.

To the above aim we propose a framework that exploits the mechanisms of the TEQUILA (Traffic Engineering for Quality of Service in the Internet at Large Scale) architecture [13] that brings together Service Management and Traffic Engineering functionality for QoS support in next generation IP (Internet Protocol) Networks.

TEQUILA uses policy based management as the key enabler for controlling system behavior. The Service Management part of the TEQUILA architecture has two objectives: to control the traffic entering the network and to commit with the service provider's QoS guarantees.

The Traffic Engineering functionality of the TEQUILA architecture is concerned with the management of physical network resources. An off-line dimensioning process is responsible for mapping the predicted traffic demand to the physical network resources. In addition, real-time operations are implemented as the means to first, balance the load amongst the established Label Switched Paths (LSPs) in the network, and second, to ensure that link capacities are appropriately distributed among the different Per-Hop-Behaviors (PHBs) sharing each link in the core network.

In the above context our proposed approach dynamically triggers the most suitable policies for the centralized off-line dimensioning process devoted to maximize business value [1], [4]. Specifically, we propose an overlay to dynamically enforce the most appropriate admission control settings for service subscriptions and service invocations. Similarly, our proposed overlay approach dynamically enforces the most appropriate preventive and corrective actions to cope with random network variability affecting a given set of

predefined business indicators [4], [7]. In addition, our overlay approach dynamically evaluates these business indicators.

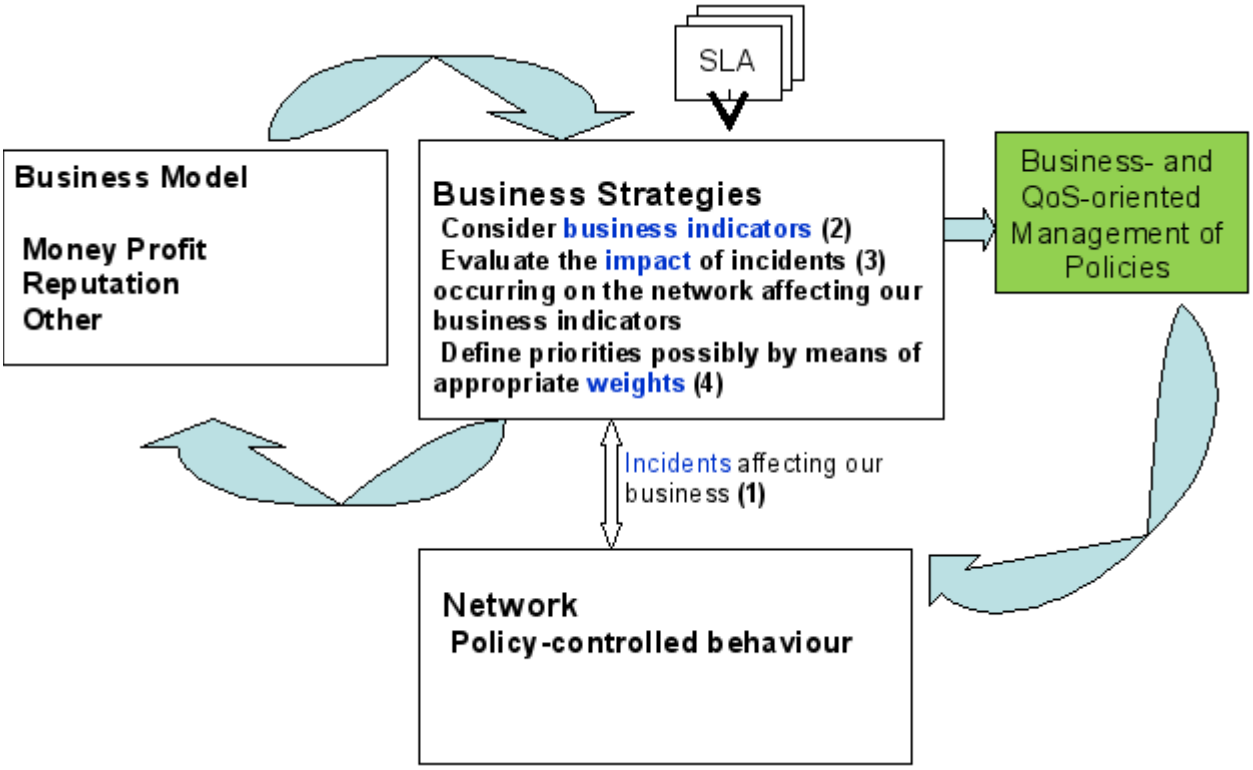


Figure 34: PRIPOL Approach: High-level View

The mechanism to derive enforceable policies from high level policies is known as policy refinement. So far, the policy refinement process in this particular application domain [18] has been carried out taking into account pure QoS-oriented aspects, isolated completely from any business considerations. In summary our system concept introduces a business optimization overlay that dynamically activates the most appropriate set of service management and traffic engineering enforceable policies as a result of a dynamic evaluation of the business indicators.

Figure 34 is a representative summary of this approach. On the left we represent the service provider business model that will be specified in terms of profit, reputation and other business concepts. In the middle, we represent the business strategies that our provider is adopting inspired in the above business model. Our claim here is that this strategy can be based on our business indicators, which depend on the incidents that randomly occur in the network but that are somehow amplified or attenuated due to the enforceable policies that drive the network behavior. Our final aim is to give a clue to the policy refiner (the green box on the right of the picture) on how to derive the enforceable policies such that to minimize (or maximize) the impact of incidents in a weighted average of business indicators.

6.2 Implementation

As said before, the proposed solution is an overlay environment in which business strategies are formalized through measurable business indicators. The idea is to priorities active changes in the managed network in order to prevent negative effects on the business indicators due to network statistical changes. These active changes are enforced through policies. More specifically, the solution consists in a multi-thread environment that collects

information from the managed network, analyses it against the business indicators and enforces changes in the active policies as to maximise the business value due to actual states of the network threatening business value.

Consider for instance a business indicator linked to “the profit generated from a certain type of customers”. The system will produce appropriate sets of policies devoted to maximize the profit obtained from that type of customers under conventional network utilisation conditions, taking into account QoS constraints. Thus, the goal is to handle several business indicators.

A critical issue in this research is the modelling that formalizes the relationships between the business indicators and pure technological aspects of the DiffServ management domain. Initially, our solution considers the identification of potential incidents affecting the business value. In the DiffServ (Differentiated Services) domain incidents may be simple ones like a “service rate threshold crossing”, or composed like “a service rate threshold crossing under network congestion state”.

A model establishes formal relationships to link simple or composed incidents with business value threatening. For example, the fact that the network is congested or in normal state may have a measurable impact on service level degradation. Also, traffic injection threshold crossings may imply that users may be starving the network, under-utilizing it or injecting traffic according to the pre-signed SLA (Service Level Agreement). In this context the modelling formalizes the effects of an incident up to the SLAs. For instance, an incident may affect a set of PHBs in a network link. Each PHB affected by an incident may in turn affect a set of services and finally, a service may have some effect on a given SLA. SLAs may eventually be linked with user information or with other service provider's information. Incident effect trees are the source of information to carry out the analysis and consequently this information should be modelled properly.

An analysis stage evaluates the total impact of incidents occurring on the network affecting all the business indicators and defines priority actions enforced with QoS policies. The central part of the analysis deals with the formalization of the effect, from now on referred to as impact, of incidents over the business indicators. Then, a policy refinement approach [18] deploys QoS-oriented policies for this last enforcement step.

The complexity of the analysis stage of the system lies in the generation of policies aimed at minimizing the negative effect of the above incidents on the business indicators throughout system execution. The total impact of an incident i should consider all the business indicators affected by such incident. As an incident may have different impact on different business indicators, the chosen approach considers the formalization of this situation by means of weights. Weights represent the degree of importance that an incident i has on a business indicator j.

The evaluation of priority actions is by no means a trivial task. The DiffServ application domain is a multi-service environment with shared resources. Priority actions committed to some user's (or administrative domains) may have some influence on other business indicators, most probably linked to other users of the network. A trade-off processing step committed to stability is mandatory at this stage of this work. As priority actions are enforced through policy, stability passes through the management cycle of policies. An event handler enables the communication amongst the monitoring sub-systems of the managed network and the overlay system itself. This way the incidents affecting our business indicators are correlated dynamically and in case of further priority actions are needed those can be effectively scheduled.

6.3 Evaluations

We are currently evaluating the concept in the DiffServ Network Management application domain under different patterns of resources utilization, patterns of traffic exchange between administrative domains and diverse network topologies. The network conditions are simulated by means of OPNET network simulator. The results of this research can be further validated in real life scenarios in which traffic traces collected from real-life capturing process may be used to prove the effectiveness of the patterns induced in this holistic study.

6.4 Modeling and Implementation Achievements and Key Results

As said above we don't have yet conclusive results but we are in the process. Our first milestone will be to have a testbed for evaluation through simulation. This testbed will consist of a Diffserv with MPLS network controlled by policies. These policies will be in charge of controlling the access to the network and we also plan to have policies entrusted to determine the LSPs between ingress and egress routers. In addition, the network will be able to create incidents like for instance the congestion of specific paths or ingress nodes.

The second milestones will consist of defining business indicators measurable in the above mentioned network. One of these candidate indicators will be the loss of revenue due to the lack of fulfillment of SLA. We have to insist in the fact that these business indicators will be able to be monitored in the whole network. Once completed the first and the second milestones we will be able to create incidents and observe the impact of these incidents in the monitored business indicators.

The third milestone will be to establish the correlation between the network enforceable policies and business indicators. A way to proceed here is for example to reproduce the same incident for different sets of QoS policies, observe the results and deduce trends that can be used at the policy refinement process. As a simple example, let's imagine we play with three sets of policies namely A, B and C to for access control. These sets differ between themselves in the value of thresholds they use to accept incoming traffic. Applying the sets in the order A, B and C we observe that our business indicator “loss of revenue due to the lack of fulfillment of SLA” shows a minimum for set B. The lesson learned would be that by properly selecting the thresholds in the admission control policies we would be in position to minimize revenue losses. This conclusion would then be informed to the policy refinement authority that eventually would make use of it in the refinement process.

Finally, we are also aiming at cases where several business indicators and several incidents will be considered at a time. In these cases we will weight the importance of the business indicators, namely a weighted average of several business indicators.

7 GridAcc: Grid Accounting

Large-scale service-oriented computing in a local computing center context of a computational Grid or across multiple domains within dynamic VOs (Virtual Organization) clearly states the need for the suitable economic management mechanisms to be in place. Accounting of services provided and resources used constitutes a management mechanism of key importance. This is due to the fact that accounting records form the basis for further related, time- and content-wise dependent management tasks. In particular, this includes charging and billing as well as analysis and optimization.

Accordingly, existing accounting approaches applicable to the specific needs of computing centers and Grid systems were analyzed and compared in the course of GridAcc. This comprehensive related work study has revealed that issues of technical precision and project-driven optimizations had been focused so far, while neither approach showed sufficient support of multi-domain or virtualization concepts. In addition, neither approach funded on the suitable accounting principles from business domains. Consequently, these identified gaps led to the development of an accounting model suitable for computing centers and dynamic VOs. This highly flexible model [15] combines principles of technical and business accounting by means of Activity-based Costing (ABC) service constituent parts and defined accountable units which take characteristics of a considered technical resource into account.

Driven by a successful preliminary functional evaluation of the developed accounting model, a full-fledged model application and evaluation case was conducted in the second phase of GridAcc. To this aim, the model was applied to the context of the Leibniz Supercomputing Centre (LRZ) [17] in Garching, Germany, lead by Prof. Dr. Heinz-Gerd Hegering being a member of LMU (Ludwig Maximilians University of Munich, Germany). Accordingly, the applicable in-depth methodology to apply and evaluate the accounting model was determined as summarized in Section 7.2 was determined. Section 7.3 outlines the respective results gained from model application and Section 7.4 focuses on those key findings identified from the performed LRZ accounting model application case. While these mentioned sections give a summary of the respective system design, implementation, and evaluation characteristics, full details are documented in the paper “Evaluation of an Accounting Model for Dynamic Virtual Organizations” [32] as published in the Journal of Grid Computing (see Section 18.2 for abstract text and table of contents).

7.1 Final System Design

The accounting model application and evaluation conducted for the respective context of the LRZ follows an elaborate methodology as documented in Figure 35. This procedure is highly complex as it has been designed for an equally complex application environment, the LRZ. It is, however, by no means limited to LRZ specifics so that the developed methodology is generally applicable irrespective of a currently considered application context.

The accounting model was designed with integration of technical and business accounting in mind. While technical accounting issues are abstracted in Figure 35, the respective core principles of business accounting in use are directly reflected. The accounting model results in a cost calculation (1) for a given service provided in a computing center. This means that by applying the model, incurred costs for a service are calculated. From an economic point of view, such a service is considered a product. A given service may embrace sub-services potentially resulting in a service tree. Cost may be calculated for any (sub-)service within a service tree (F). It is important to note that cost calculation in this context relates to a full cost calculation. Accordingly, calculated full costs represent a longer term cost perspective.

Any cost calculation relies on suitable input data. Input data is gained by the help of two established business accounting methods. The first is TCAS (Traditional Cost Accounting System) which is time-wise preceding the second, ABC (Activity-based Costing). TCAS principles are applied in blocks (A), (B), and (C), while ABC is reflected by blocks (D) and (E). This separation of TCAS and ABC was introduced to combine strengths of both business accounting approaches. TCAS is used for calculation input data in relation to annual cost elements that are gained from a typical annual balance sheet. These annual

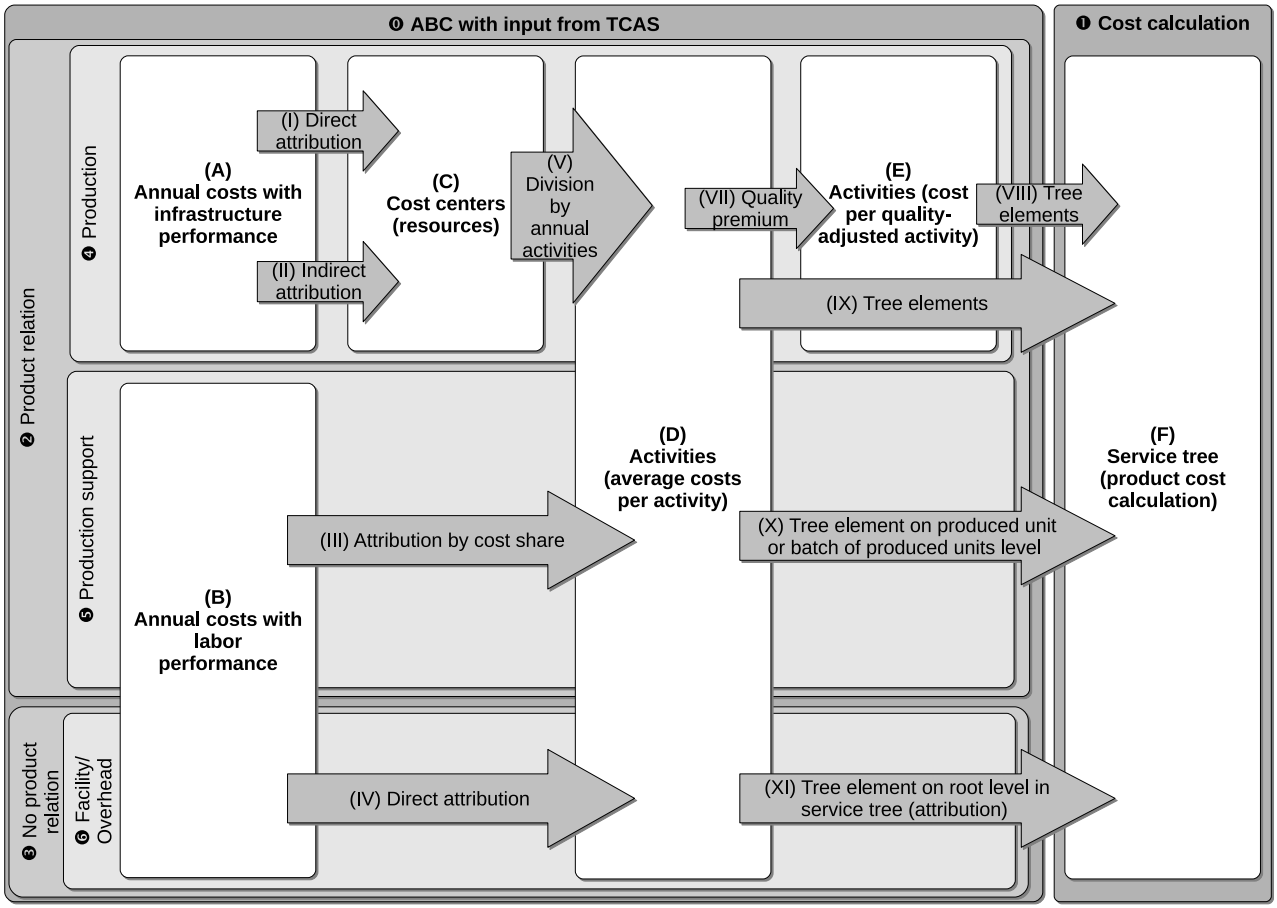


Figure 35: GridAcc Model Application Methodology [32]

cost input elements are structured according to whether they show a direct relation to the end product (2), i.e. to that service for which costs are finally calculated (F), or whether they lack such relation to the product (3). Input data is further structured according to any given relation to production (4), production support (5), or facility management and overhead (6). (4) covers annual costs with infrastructure performance such (A) such as annual depreciation on buildings or annual energy cost that are incurred by the operation of a computing center. Accordingly, (5) and (6) cover annual costs with labor performance (B) like working hours spent in IT service management as an example for (5) and administrative work as an example for (6).

Depending on the applicable type of costs gained from TCAS, these costs are directly (I) or indirectly (II) attributed to cost centers (C) — cost centers reflect deployed IT resources in a computing center here —, or, in case such intermediary cost center attribution is not required, input costs are attributed to activities (D). Activities are at the very core of the developed accounting model. This is the point where ABC principles are applied. ABC seeks to determine costs for activities — instead of determining costs for cost centers and cost objects, as it is the case in TCAS. From a TCAS perspective, activities appear as cost objects. Thus, TCAS and ABC are inter-linked by means of activities. TCAS delivers average annual costs which the respective attributed activities then need to cover. ABC has proven to be the business accounting principle of choice for service businesses. ABC, thus, satisfies the requirements of a computing center best.

Activity-based accounting in the developed model is facilitated by means of so-called service constituent parts (D) of types *Processing*, *Storage*, *Transferring*, *Output*, *External*, and *Other*. Service constituent parts are bound to the respective applicable accountable

units. For instance, processing-related service constituent parts are accounted based on how many CPU seconds and how much main memory a (sub-)service consumes. Service constituent parts, thus, are configurable to the respective reflected IT resource(s). The accounting model, furthermore, assumes a standard configuration for any considered (sub-)service. In case such a standard configuration needs to be adapted (E) to the needs of a customer, a premium is added in respect to additional service and configuration management effort incurred. All services and sub-services of the resulting product cost calculation (F) are composed from the respective accounted service constituent parts *Processing*, *Storage*, *Transferring*, *Output*, *External*, and *Other*.

7.2 Implementation

Driven by the developed methodology as summarized in Section 7.1, the accounting model was applied to the environment of the LRZ. In-depth scenario description, IT resource considerations, core implementation design principles, and an embracing application results presentation can be found in [32], whereas focus is set here on covering the implemented full cost calculation.

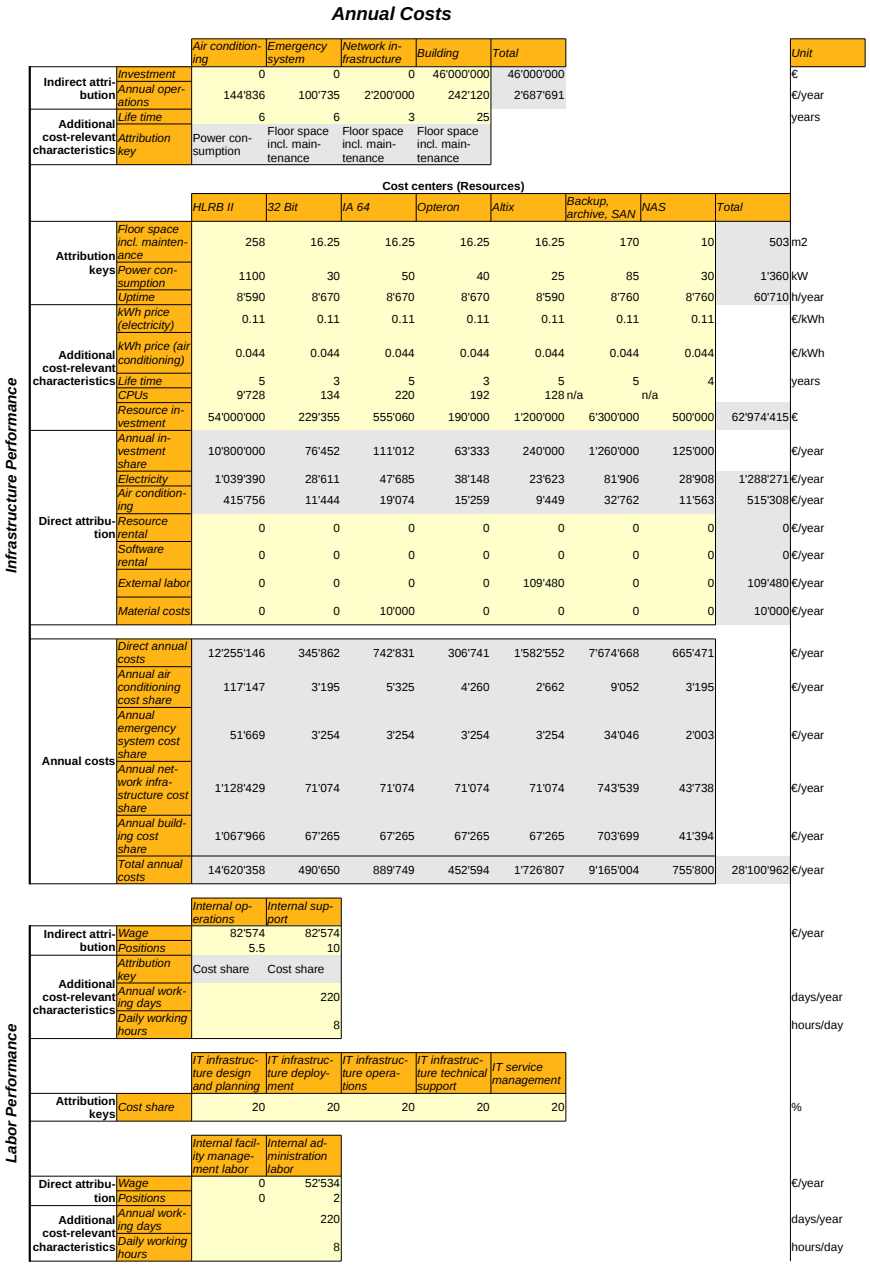
The implementation involves three main elements which directly reflect and instantiate the previously outlined system design (cf. Figure 35). These three elements involve on one hand annual cost calculations from TCAS and from ABC. While the first content-wise representation represents blocks (A) to (C) of Figure 35, the latter implements blocks (D) and (E) of Figure 35. On the other hand, the third implementation element covers the actual product cost calculation of the respective service tree as envisaged by block (F) of Figure 35.

The accordingly resulting calculation shown in Figure 38 was conducted according to a previously determined scenario. Even though this scenario — depicting a multi-domain, computationally and storage-intensive Grid service for complex physics simulation and results visualization — models a service which does currently not exist at the LRZ, the scenario bases on LRZ-specific input data and constitutes a full accounting model application case. The calculation for the considered scenario results in product costs of 4656 EUR. This means that costs of 4656 EUR are incurred every time this service is invoked and successfully completed.

7.3 Evaluations

Evaluation of the accounting model was performed with respect to three dimensions:

- Model functionality: The model was assessed with regard to its principle functionality, meaning what the model was able to deliver in terms of results and what input data was required to get to these results. In particular, the expressiveness of gained results was analyzed which included a listing of achieved insight and encountered issues or limitations.
- Model parametrization: Since the accounting model was designed with maximum flexibility and support of multiple accountable units in mind, the respective set of used parameters was of core interest to evaluate the model's characteristics. Metrics and model parameters were assessed in both, static as well as dynamic manner, the latter including multiple sensitivity analyses.



Annual Costs

		Air conditioning	Emergency system	Network infrastructure	Building	Total	Unit
Indirect attribution	Investment	0	0	0	46'000'000	46'000'000	€
	Annual operations	144'836	100'735	2'200'000	242'120	2'687'691	€/year
Additional cost-relevant characteristics	Life time	6	6	3	25		years
	Attribution key	Power consumption	Floor space incl. maintenance	Floor space incl. maintenance	Floor space incl. maintenance		

Cost centers (Resources)

		HLRB II	32 Bit	IA 64	Opteron	Altix	Backup, archive, SAN	NAS	Total	
Infrastructure Performance										
Attribution keys	Floor space incl. maintenance	258	16.25	16.25	16.25	16.25	170	10	503	m2
	Power consumption	1100	30	50	40	25	85	30	1'360	kW
	Uptime	8'590	8'670	8'670	8'670	8'590	8'760	8'760	60'710	h/year
Additional cost-relevant characteristics	kWh price (electricity)	0.11	0.11	0.11	0.11	0.11	0.11	0.11		€/kWh
	kWh price (air conditioning)	0.044	0.044	0.044	0.044	0.044	0.044	0.044		€/kWh
	Life time	5	3	5	3	5	5	4		years
	CPUs	9'728	134	220	192	128	n/a	n/a		
	Resource investment	54'000'000	229'355	555'060	190'000	1'200'000	6'300'000	500'000	62'974'415	€
Direct attribution	Annual investment share	10'800'000	76'452	111'012	63'333	240'000	1'260'000	125'000		€/year
	Electricity	1'039'390	28'611	47'685	38'148	23'623	81'906	28'908	1'288'271	€/year
	Air conditioning	415'756	11'444	19'074	15'259	9'449	32'762	11'563	515'308	€/year
	Resource rental	0	0	0	0	0	0	0	0	€/year
	Software rental	0	0	0	0	0	0	0	0	€/year
	External labor	0	0	0	0	109'480	0	0	109'480	€/year
	Material costs	0	0	10'000	0	0	0	0	10'000	€/year
Annual costs	Direct annual costs	12'255'146	345'862	742'831	306'741	1'582'552	7'674'668	665'471		€/year
	Annual air conditioning cost share	117'147	3'195	5'325	4'260	2'662	9'052	3'195		€/year
	Annual emergency system cost share	51'669	3'254	3'254	3'254	3'254	34'046	2'003		€/year
	Annual network infrastructure cost share	1'128'429	71'074	71'074	71'074	71'074	743'539	43'738		€/year
	Annual building cost share	1'067'966	67'265	67'265	67'265	67'265	703'699	41'394		€/year
	Total annual costs	14'620'358	490'650	889'749	452'594	1'726'807	9'165'004	755'800	28'100'962	€/year

		Internal operations	Internal support	
Labor Performance				
Indirect attribution	Wage	82'574	82'574	€/year
	Positions	5.5	10	
Additional cost-relevant characteristics	Attribution key	Cost share	Cost share	
	Annual working days		220	days/year
	Daily working hours		8	hours/day

		IT infrastructure design and planning	IT infrastructure deployment	IT infrastructure operations	IT infrastructure technical support	IT service management	
Attribution keys	Cost share	20	20	20	20	20	%

		Internal facility management labor	Internal administration labor	
Direct attribution	Wage	0	52'534	€/year
	Positions	0	2	
Additional cost-relevant characteristics	Annual working days		220	days/year
	Daily working hours		8	hours/day

Figure 36: Annual Input Costs from TCAS [32]

- Model application context: The accounting model was (for the first time) applied to a complex environment, namely the LRZ, using a comprehensive scenario. Accordingly, available and used input data from LRZ as well as key determinations of the scenario considered were assessed. Furthermore, sensitivity analyses for scenario parameter changes were conducted.

7.4 Modeling Achievements and Key Results

Driven by those three evaluation dimensions introduced and shortly described in Section 7.3, the model application and evaluation case to the LRZ environment and a multi-domain Grid scenario was, in general, found to be highly successful. Nevertheless, sufficient room for model extensions and future work was found due to some limitations encountered. While full details are available in [32], those detailed evaluation findings are summarized are summarized subsequently.

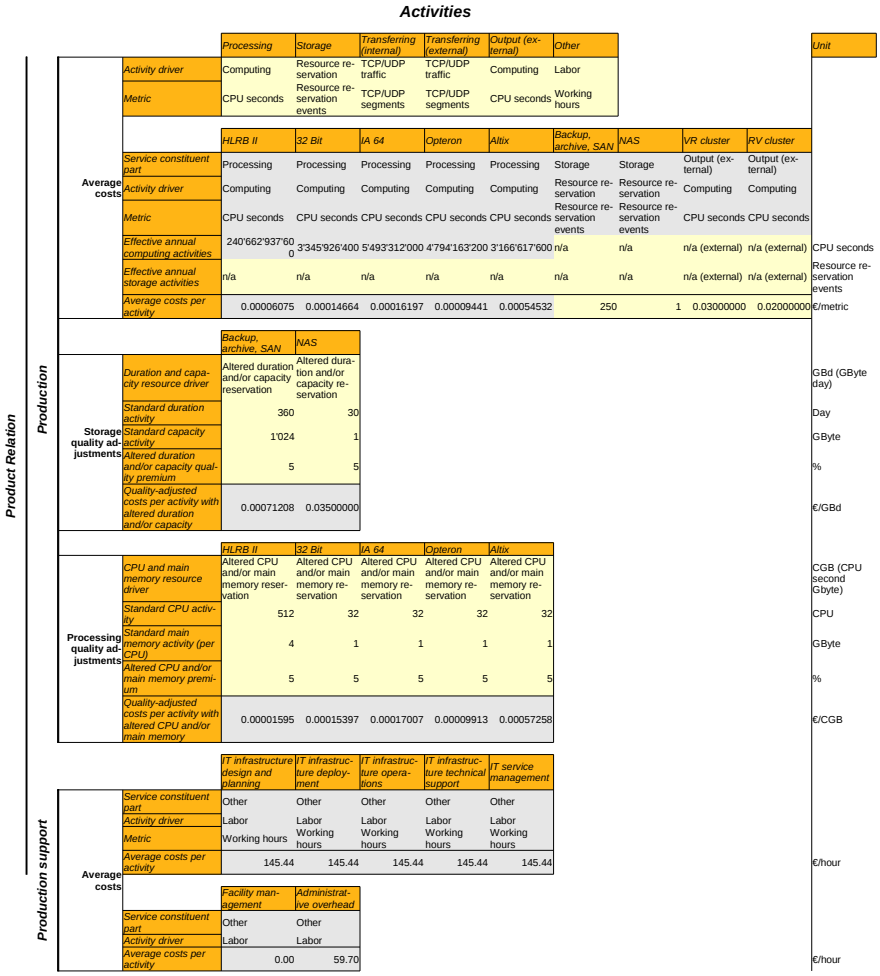


Figure 37: ABC Activities Calculation [32]

With respect to the model's functionality, the model was found to expose a high level of expressiveness. The main goal of a full cost calculation incorporating principles of technical and business accounting was achieved for an application case featuring real-world LRZ input data and a fictitious but still realistic multi-domain Grid service provisioning scenario. The applied principle of resource-adapted, quality-aware service constituent parts was assessed particularly helpful in cost calculation, since the introduced method of standard- and adaptable configuration reflects a user demand which is seen quite commonly in this application context. Furthermore, the model was found to cope successfully with all requirements set on a service tree down to the accordingly matched service constituent parts. This means that the model has proven to be self-contained in terms of configuring those standard service constituent parts of types processing, storage, transfer, and other. Moreover, the model was even able to consider input lacking input data in the area of Grid traffic. This data, which was supposed to be metered within the application environment, was not available. Only annual raw network traffic costs from annual balance sheets was available. Due to the accounting model's inherent use of ABC and TCAS, the respective network traffic costs could still be considered as TCAS input data.

Similar to these positive findings in the area of model functionality, the accounting model was found to be highly flexible with regard to parametrization possibilities. However, a number of model parameters had to be estimated from previous LRZ experience or had to be assumed from comparable businesses. In that light, sensitivity analyses conducted determined a valuable means to identify effects of parameter changes to product costs.

IT Product Cost Calculation

Simulation

		Accounted CPU seconds per CPU	CPUs	Main memory per CPU	Activity costs	Unit
Processing	HLRB II	9'000	512	2	146.97	€
	IA 64	14'400	220	1	538.78	€
	Altix	32'600	32	1.5	895.98	€
		Duration	Capacity	Activity costs		
Storage	NAS	5	2'048	358		€
	Backup, archive, SAN	360	5'120	1'313		€
		Billed CPU seconds	Activity costs			
Output (external)	VR cluster	3'600	108.00			€
	RV cluster	3'600	72.00			€
		Accounted working hours	Unit/batch activity mapping factor	Activity costs		
Other	IT infrastructure design and planning	10	0.05	73		€
	IT infrastructure deployment	10	0.20	291		€
	IT infrastructure operations	10	0.20	291		€
	IT infrastructure technical support	5	0.10	73		€
	IT service management (standard)	30	0.05	218		€
	IT service management (quality-adjustments)	1	1.00	145		€
	IT service management (continuity management)	0	1.00	73		€
		Accounted working hours	Activity costs			
Other	Facility management	0.5	0.00			€
	Administrative overhead	1	59.70			€
Product costs	4'655.86					€

Figure 38: Product Cost Calculation [32]

Figure 39 shows the accordingly conducted sensitivity analysis for a number of key input parameters. These parameters were varied by 10%, while all other parameters were kept at the initial level. Table fields show the respective relative change in calculated product costs caused by such a parameter change. The top five impact classes have been determined as follows:

- Product cost changes in the range of 2.6-2.8% (a leverage of around a fourth) result from 10% changes in Backup, archive, and SAN parameters each.
- Product cost changes in the range of 1.5-1.7% result from 10% changes in selected Altix cluster and internal support labor wage or position parameters each.
- Product cost changes in the range of 0.9-1.1% result from 10% changes in selected IA 64 cluster and internal operations labor parameters each.
- Product cost changes in the range of 0.7-0.8% result from 10% changes in selected NAS infrastructure parameters each.

With regard to application context assessment, comparable sensitivity analyses were conducted. These, however, assessed parameter changes of 10% and their respective results to product costs for parameters relevant to the chosen application scenario. These relative changes were of interest since the scenario itself was deemed artificial to some extent, as the LRZ does not currently offer a multi-domain Grid service like the one comprised in the scenario. The respective top five identified impact classes range from 2.8% (Backup, archive, SAN scenario parameters), 1.9% (Altix cluster), 1.2% (IA 64 cluster), 0.9% (IT service management) to 0.8% (NAS duration and capacity).

Calculation Input Parameter Sensitivity Analysis

	Building	Air conditioning	Emergency system	Network infrastructure
Investment	0.2	<0.1	<0.1	0.2
Annual operations	<0.1			
Life time	0.1			

	HLRB II	IA 64	Altix	Backup, archive, SAN	NAS	VR cluster	RV cluster
Floor space	0.1	0.2	0.1	0.1	<0.1		
Power consumption	<0.1	<0.1	<0.1	<0.1	<0.1		
Uptime	0.3	1	1.7	<0.1	<0.1		
kWh price (electricity)	<0.1	<0.1	<0.1	<0.1	<0.1		
kWh price (air conditioning)	<0.1	<0.1	<0.1	<0.1	<0.1		
Life time	0.2	0.1	0.2	<0.1	<0.1		
CPUs	0.3	1.1	1.7				
Resource investment	0.2	0.9	1.6	<0.1	<0.1		
Resource utilization	0.3	1.1	1.7				
Average costs per activity				2.8	0.8	0.2	0.2
Standard main memory activity	0.3	1.1	1.7				
Standard duration activity				2.6	0.7		
Standard capacity activity				2.6	0.7		
Altered duration and/or capacity quality premium	<0.1	<0.1	0.1	0.1	<0.1		

	Internal operations	Internal support	Internal administration labor
Wage	0.9	1.6	0.1
Positions	0.9	1.6	0.1

	IT infrastructure design and planning	IT infrastructure deployment	IT infrastructure operations	IT infrastructure technical support	IT service management
Cost share	0.4	0.2	0.2	0.4	0.5

Figure 39: Input Parameter Sensitivity Analysis [32]

Scenario Parameter Sensitivity Analysis

	HLRB II	IA 64	Altix	Backup, archive, SAN	NAS	VR cluster	RV cluster
Accounted CPU seconds per CPU	0.3	1.2	1.9				
CPUs	0.3	1.2	1.9				
Main memory per CPU	0.3	1.2	1.9				
Duration				2.8	0.8		
Capacity				2.8	0.8		
Billed CPU seconds						0.2	0.2

	IT infrastructure design and planning	IT infrastructure deployment	IT infrastructure operations	IT infrastructure technical support	IT service management	Administrative overhead
Accounted working hours	0.2	0.6	0.6	0.2	0.9	0.1
Unit/batch activity mapping factor	0.2	0.6	0.6	0.2	0.9	0.1

Figure 40: Product Cost Sensitivity Analysis [32]

Overall specific areas for future model extensions and enhancements were found [32]:

- Consideration of load balancing aspects.
- Extension of the concept of quality premiums to better support competition for resources.
- Consideration of costs caused by unused but not attributable resources in a more fine-granular way.
- Definition and integration of generally applicable set of metering points for technical accounting.

8 Overall Economic Management Model

While the preceding sections do discuss those important models, approaches, and solutions separately, the question remains, in which way to they form a key part of an overall economic management model for network traffic in the Internet. This question is answered below by applying the Deployment model (cf. Figure 2) to each of those separate steps and placing them into relation to each other.

The ***ASAM approach*** monitors SLAs in a multi-domain network. It thus considers the aspect that multiple providers need to cooperate in order to provide a service - here the transporting of IP packets from a sender to a receiver - but also the aspect, that those providers are independent business entities and sometime even business rivals. The later is taken care of by protecting the privacy of a provider and by determining the location, where in the network, i.e., in which Autonomous System, a violation happens.

In this respect ASAM is one link of the value chain, by providing a feedback to the client, how well a provider provided a service and thus what the client actually received for his money. This paves the way for new services and a new profit potential, where a provider can offer premium services for a higher price, which the client pays, because he can measure the service quality and together with a respective incentive scheme can actually force the provider to comply with the agreed service levels.

Clear relationships to the other projects presented in this deliverable and the ASAM project can be identified. The ***SaPDoGS approach*** can help designing SLAs. The promise theory might help creating “better” SLAs, which would be a competitive advantage. Improving processes and procedures, as done by BP3EM project, assists providers operating their networks more efficiently, thus complying with SLA or recover after SLA violations quicker. Finally the PRIPOL project develops new price model, which are necessary when auditing should generate additional revenue for providers.

The SAPDoGS project attempts to look with fresh eyes at the issues of expectation of Service Levels in distributed systems. Distributed services are difficult to measure and evaluate because they have no common standard. The brokering and scheduling of several jobs is a common scenario in Grid computing, required by a variety of applications. Promises allow for dynamically and efficiently matching flexible user requirements with the availability and dependability of Grid resources, through the negotiation and re-negotiation of promises. Promise theory predicts that the main reason for centralization of observation is the need for low-cost distributed calibration.

Grids are only one example of distributed systems but they serve as a case where performance guarantees have been one of the motivating factors for developing the techniques. Promise theory offers a system-neutral approach to understanding the functional and economic relationships between components in any distributed system. This paradigm enables users and system software to successfully schedule and orchestrate complex computational Grid jobs across multiple domains. Thus a Grid accounting Model with a full cost calculation is required. Monitoring and accountability are becoming increasingly important in networked environment. Therefore commercial Grids need to develop appropriate monitoring and metering tools for observing resource utilization managing performance degradations, availability and other parameters. Observed data needs to be validated against commitments made to consumers. Automation of these processes is highly desirable due to the scale and complexity involved. Thus, the complementary relation to GridAcc is laid out with regard to these mentioned aspects.

In order to formulate realistic and effective promises to manage these issues, the need for models on service delivery are obvious and the experience of “best practices” such as those discussed in BP3EM. Indeed, BP3EM examines a technology for encoding promises directly as a driver of service delivery, using cfengine – the principal tool of promise theory.

The auditing of data exchanged between domains is one way to verify compliance with promises. ASAM effectively contributes to a discussion of how economics may be impacted by observation of behavior. According to promise theory this can only be an estimate since the full economic picture also depends on intangibles of the relationships between parties, such as reputation which cannot be directly observed. Promises are a basic model for expressing policies between independent domains. The modification of a policy at run time —as investigated in PRIPOL— is something that can only be modeled using an autonomous viewpoint. Promises allow the detection of conflicts in distributed systems, in a way that is extremely cumbersome using other models. Promise theory requires a reformulation of the problem but this has many advantages over other approaches.

As a part of the overall Economic Management model, the ***BP3EM approach*** is subject to the second segment of scope - Optimization of IT Service Management - as depicted in Figure 1. It completes and extends work being done in previous stages of this EMANICS work package. The effort of integrating the most popular process framework for IT Service Management, the IT Infrastructure Library (ITIL), with one of the widest-spread policy/ promise-based Configuration Management tools, cfengine, provides an exemplary, but thus practical and significant contribution to the goals and vision of the entire project. It is a piece of proof that as well as practical guidance on *how* the organizational dimension and the technical view point of service management and system administration can be better aligned. For service providers facing the challenges of future internet management, these results can be of great value, although many difficulties and challenges remain and demand for further analysis, research and implementation efforts, as addressed in detail in Section 5.3

BP3EM focuses — due to the complexity it brings anyway — on single domain scenarios and, thus, does not address multi-domain aspects. With respect to the mechanism’s scope segment of the Economic Management model, BP3EM — in partial contrast to the other projects — covers not only the technical, but also the organizational dimension of (economic) IT management.

The ***PRIPOL approach*** establishes mechanisms to determine the effect of policies on business value and therefore aiming at policies that get the maximum value. As such, it covers the area of Optimization of IT Service Management but it also covers Multi-domain aspects in the sense that these mechanisms can also be applied in multi-domain scenarios.

PRIPOL addresses the market of IT service providers relaying on a network infrastructure belonging to them or to a third party. Therefore the value network is constituted by the service providers, network providers and service customers. The service model adopted in PRIPOL is grounded on the proposition that policies, that is network enforceable policies derived from high level service policies, are impacting the business value of the service offered to service customers in addition to their primary aim that can be the access control, the QoS assurance, etc. If this business value is quantified, though a given set of “business indicators”, the service provider will be able to control these policies to maximize the former and, ultimately, to design policies to this specific aim.

PRIPOL is not bundled to any specific charging model because it can be adapted to any. We could say that the PRIPOL will lead to a business value driven pricing scheme in the sense that the ultimate goal is to find pricing policies that maximize the business value of

the service. The competitive strategy here is that services providers adopting PRIPOL mechanisms will differentiate from others with the same services and potential customers because they will get better business results than competitors.

PRIPOL can be seen at a higher level than at least three projects described in this deliverable and therefore it can self-support on them. In fact, results from ASAM can benefit PRIPOL helping to determine if SLAs are really fulfilled or not by the policies that are intended to do so. On he other hand, it is crucial for PRIPOL to have powerful and efficient accounting mechanisms in order to determine the key indexes that are needed in its approach, namely the business indicators. This can be provided by the project GridAcc. Also, PRIPOL makes use of the term “incident” that is aligned with the same term in ITIL. Therefore as ITIL provides guidance for incident management and BP3EM studies the use of Cfengine to support ITIL we believe that PRIPOL could also take advantage of this application. Finally, the SaPDoGS project defines a paradigm for service assurance based on promise theory as an alternative to SLAs. Although PRIPOL makes use of the concept of SLA we believe that it would be also possible to express it in terms of promises.

The ***GridAcc approach*** is positioned in the overall economic management model as an key driver for optimizations of IT service management in computing centers by developing the detailed economic management mechanisms for an optimal accounting in Grid systems. GridAcc is concerned with the design and implementation of an accounting model that allows to perform a full cost calculation for a Grid service. This cost calculation leverages established business accounting principles and it integrates these with accountable units from technical accounting.

GridAcc addresses the market for large computing centers, involving a value network composed from actors such as Grid service providers, Grid infrastructure providers, network providers, Grid service users etc. GriAcc bases on an elaborate service model decomposing business processes into atomic Grid service constituent parts of processing, storage, transferring etc. which are resource-specific, allow for QoS, and all carry the appropriate cost attribution. The underlying service model supports Virtual Organizations and cost attribution leverages established cost accounting principles, proven to fit best with requirements of cost accounting in service businesses. GridAcc does not require the adoption of any charging model, but it supports most directly a cost-driven pricing scheme due to its inherent focus on cost calculation. GridAcc helps operators of computing centers to identify costs incurred by a specific service provided and to identify potential inefficiencies in their cost structure. Therefore, GridAcc may serve as a means to increase competitiveness.

GridAcc contributes to the overall economic management model in several ways. As it adopts the perspective of Grid service providers and large computing center operators, focus is set on a full cost calculation and cost-driven optimization aspects with respect to provided services. A full cost calculation may serve as the reliable basis for a corresponding (cost-driven) pricing strategy — results from GridAcc, thus, may be taken as input to determine optimal pricing policies as envisaged by PRIPOL. Similarly, knowledge about full costs that a service causes and insight into cost-related inefficiencies in service provisioning is of direct importance to the optimization of IT service management. BP3EM, thus, may benefit from GridAcc results. The same holds true for SaPDoGS, since GridAcc enables a Grid service provider in a most flexible and universal way to develop — from a costs point of view — a better understanding of what business promises may be feasible. Furthermore, GridAcc completes the overall economic management model by the respective provider perspective which is complementary to ASAM's user-driven viewpoint.

9 Summary, Conclusions, and Next Steps

The modeling of diverse economic principles in a network management context has been addressed in work package WP8 of EMANICS. This approach undertaken has extended traditional basics of purely technical network management work and integrated basic economic principles as a leading set of ideas to develop and investigate a new range of mechanisms. This is in the process to lead to a generic information infrastructure enabling multiple parties' services access, *e.g.*, between multiple domains, in an economically maintained manner. Therefore, the economic dimension of network management for Internet Service Providers (ISP) has become a reality for IP-based network solutions, which are run and offered in a commercial environment. Thus, this work undertaken in WP8 does provide an initial, small but focussed insight into the main and side effects of this economically/technically motivated situation. The partial underestimation of economic influences in an operational environment has been proven wrong, and a.o. auditing, Service Level Agreements, policies, and best practices do help to start an improving understanding as well as solution proposal for such questions.

Therefore, the EMANICS WP8 on “Economic Management” decided to investigate a dedicated number of important aspects of such economically-driven network management and network operations-based mechanisms. These include

- Auditing for Service Level Objectives (SLO) across provider domains [ASAM],
- Service Level Agreement (SLA) and promise descriptions for Grid services [SaPDoGS],
- Best practices, processes, and promises in economic management [BP3EM],
- Pricing by policies [PRIPOL], and
- Grid accounting [GridAcc].

Concluding, the selected result of those investigated projects shows that issues of economic management have been tackled in an integrated manner. Table 10 below reflects this by a comparison according to key characteristics of a well maintained economic model going beyond current state. These dimensions include the particular approach adopted (applied methods) and the respectively addressed management function. Furthermore, Table 10 highlights which management instrument results from each project and what domain(s) a project is targeting. This comparison reflects selected, nonetheless integrated, work efforts in an overall economic management model by means of a strong emphasis on challenges which become apparent and needed in a commercial context. Issues of auditing, pricing, and cost accounting constitute core management functions in this overall economic model which, so far, typically remained partly neglected or fully abstracted by established network management model like the FCAPS (Fault, Configuration, Accounting, Performance, and Security) model.

Auditing, for instance, is considered in FCAPS from a security viewpoint, while in this new economic management model the respective range of auditing functions determines mainly the central trust building and, thus, business sustaining mechanism. Consequently, each of those five projects newly addresses the relevant extensions required for a well maintained integrated economic management model — that is a model which facilitates a flexible policy-based pricing according to business metrics, which adds a fresh look at service management and partly automated, adaptive configuration by means of promises and cfengine policies, a model which ensures the needed integration of both, provider and user perspectives, in terms of an in-depth insight into costs incurred by a service offered as well as in terms of a reliable SLA compliance auditing.

Table 10: Aspects Covered by the Overall Economic Management Model

Project	Applied Methods	Addressed Management Function	Resulting Instrument	Addressed Administrative Domain
ASAM	SLA Auditing	Monitoring, Auditing	SLA compliance report including information where a potential violation occurred.	Multi-domain (multiple ISPs, ASes)
SaPDoGS	Promise Theory	Configuration	Methodology to apply promises on Grid services	Single-domain (organization-internal)
BP3EM	ITIL IT Service Management	Configuration	Guidelines for ITIL objective implementation in cfengine Terminology mapping between ITIL and cfengine	Single-domain (organization-internal)
PRIPOL	Policy-based Management	Pricing	Methodology to determine the market price of services based on policies through measurable business indicators	Primarily single-domain
GridAcc	Cost accounting (TCAS and ABC)	Accounting	Full cost calculation for a service composed of the respective mix of computing, storage, transferring, and output activities.	Primarily single-domain (large computing center), secondarily multi-domain (VO)

The emerging needs and public interest in economic management can be seen — besides the publications of WP8 done and summarized in Annex 5 (cf. Section 18 below) — by the number of international research and standardization events, which took place in the year of 2008. Thus, a report on a “Promise Theory Workshop” in Annex 1 (cf. Section 14) shows the interest of the public and guidance of EMANICS in that area. Furthermore, Annex 2 (cf. Section 15) reports on the “Business-driven IT Management (BDIM)” workshop, which organized discussions on the question: in which way can IT management be aligned with business and economic goals of organizations? Finally, the work on economic management has risen interest in the IETF (Internet Engineering Task Force) as well. As shown in much more detail in the Annex 3 and 4 (cf. Section 16 and Section 17 below), a WP8 partner participated in the discussion and forming process of the ALTO WG (Application-layer Traffic Optimization Working Group) during the last two IETF meetings. While the focus in this work was on P2P (peer-to-peer) traffic particularly, to determine the differences between peer-to-peer (resource-based) and Client/Server (host-based) traffic. The purpose of this meeting in Dublin was to become a WG in the IETF to define only the protocol – not caring about metrics. The Minneapolis meeting saw the formal founding of the ALTO WG. Thus, further work is expected at that end.

As discussed, the set of dedicated areas of Economic Management remains still by far not complete. Thus, a next set of aspects — based on WP8 experiences gained and partially extending the basis of D8.4 — are expected to be investigated in the last and upcoming year of WP8, while the emphasis will be laid on the optimization of existing IT service management approaches, refinements of concrete economic or economics-supporting

mechanisms, and a dedicated legal investigation of a contract's dependencies on multi-domain aspects.

Firstly, an approach following the BP3EM principles and termed SLA Planning and Negotiation (SLAPN) will investigate and develop an information as well as a functional model for the field of SLA management to provide a comprehensive, generic (i.e. applicable to all deployment scenarios of interest), and detailed model of the information objects that have to be supervised and managed in the context of Service Level Management (SLM). Thus, with respect to Figure 1 the topic 2 will be addressed.

Secondly, the extension of the Grid accounting (GridAcc) work is planned to ensure that it will cover load balancing aspects as well as costs caused by unused, but not attributable resources. The definition and integration of a generally applicable set of metering points for technical accounting will be included as well, resulting in an applicable model and respective guidelines for a computing center application. This work will address the topic 3 (cf. Figure 1).

Thirdly, the Impact Analysis of Economic and Legal Objectives on Basic and Value-added Services (IELOS) work addresses the optimization of network and service management functionality to meet economical objectives. This will be complemented with policy definitions for an automated determination of jurisdiction and applicable law according to a to-be-concluded international contract, typically done across domain boundaries and in a multi-domain manner, of an electronic value-added service. Obviously, this determines the need for any commercial settings of service provisioning in the future.

Finally, the PRIPOL approach will develop further, in which the derivation of prices for services makes use of additional policy analysis techniques, where conclusive results are to be obtained.

10 Glossary

This section outlines again the major terms, which form the basis of those key multi-provider, and SLM models as well as of those service provisioning concepts described in this document. Additionally, it has been extended by those terms of relevance from D8.2.

- **Agreement**
  A common understanding about knowledge that is shared between two parties. Agreements are often assumed to be about policies or actions (agreements to act) and are often formalized using contracts in which case both parties agree to the terms of a common contract.
- **Architecture**
  An Architecture describes interactions of components of a complex system. Often, Architectures provide a layered or comparably structured view on the respective system.
- **Contract**
  a bilateral bundle of promises between two agents, that is intended to serve as the body of an agreement.
- **Framework**
  A Framework represents a reusable design for a system by describing concepts and structures that give guidance for the execution of a system's complex tasks without

providing strict and mandatory implementation requirements or specifications. A Framework is often less concrete than a Model and described in a natural language rather than by using formal modeling techniques.

- **Model**
  A model is a representation or description designed to illustrate the structure or method of operation of an object, system or concept. In this capability, models are often used to simplify, down-scale and/or abstract from real-world entities.
- **Multi-domain**
  Adjective designating the characteristic (*e.g.*, of a management framework), that more than one administrative domain is involved. These domains often have to establish cooperation agreements on a peer to peer basis, coordinating aspects like configuration management or interaction strategies.
- **Multi-provider**
  Adjective designating the characteristic (*e.g.,* of a management framework), that more than one provider is involved. Multi-provider scenarios usually entail technical, operational and economic or legal cooperation issues between the participants to be solved by agreements.
- **Policies**
  A policy defines a course or method of action selected among alternatives and in light of given conditions to guide and determine present and future decisions.
- **Promise Agreement**
  A promise agreement is a pair of promises between two parties to acknowledge the content of an contract body.
- **Service**
  A service is the entity or unit of work offered by a service provider on behalf of a service consumer who can use the service. In general, a service includes several types of resources, including hardware- and software resources such as computing power, network links, storage capacity, and content, and it may even be composed of several sub-services.
- **Service Catalog**
  A Service Catalogue contains definitions of standard services as well as documentations of customer-specific services. It can be used as a foundation for automated service subscription or for the negotiation of SLAs.
- **Service Level Agreement (SLA)**
  A Service Level Agreement (SLA) defines the terms under which a service is offered to a service customer at a specific Service Access Point (SAP). The SLA includes a set of parameters which specify the service and the QoS under which it is provided (*e.g.*, the amount of bandwidth allocated, the involved session partners, metrics and algorithms that are used to compute SLA parameters), accountable units and the tariff which is used to charge for the service usage. Besides, several other aspects such as penalties or actions, respectively, to be taken if SLA objectives (*i.e.,* guarantees) are violated, trust relationships are part of a SLA.
- **Service Provisioning**
  IT services can be associated with a service life cycle that subsumes the steps from planning to termination of a particular service. A popular simple service life cycle is called Plan-Build-Run, typically rerun after a Change or Improvement step. Service Provisioning mainly deals with the plan, build and change parts, providing the necessary input for run time operations. It is therefore a major part of the service life cycle. More

precisely, service provisioning includes tasks such as planning new services, building the basic infrastructure, SLA negotiation and order processing, identifying adequate resources for service delivery or adapting existing services to specific customer needs, specifying steps for service implementation and service operation, up to dynamic, near real-time service composition out of service modules based on customer requirements.

- **Virtual Organization (VO)**
  A Virtual Organization (VO) is a form of organization abstraction. It is understood as a temporary or permanent coalition of geographically dispersed individuals, groups, organizational units or entire organizations that share resources, capabilities and information to achieve common objectives. VOs can provide services and thus, can take the role of a service provider.

11 References

[1] I. Aib: *A Business-driven Approach to Policy Optimization*; Doctoral Thesis, University Pierre & Marie Curie, pp. 1-140, July 2007.

[2] S. Akhil, G. Sven, M. Vijay, M. Aad: *Specifying and Monitoring Guarantees in Commercial Grids through SLA*; Technical Report HPL 2002-324, December 2002.

[3] D. Aredo, M. Burgess, S. Hagen: *A Promise Theory View on the Policies of Object Orientation and the Service Oriented Architecture*; Science of Computer Programming, 2006.

[4] C. Bartolini, M. Salle, M: *Business Driven Prioritization of Service Incidents*; Lecture Notes in Computer Science, Springer, pp. 64-75, 2004.

[5] T. Bauer, S. Bel Haj Saad: *Virtualizing Resources: Customer-oriented Cross-domain Monitoring for Service Grids*; 10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07), pages 777-780, 2007.

[6] U. Binder: *Ehevertrag für IT Dienstleistungen*; Infoweek Vol. 34(4), August 2001.

[7] M. J. Buco, R. N. Chang, L. Z. Luan, C. Ward, J. L. Wolf, P. S. Yu P: *Utility Computing SLA Management Based upon Business Objectives*; IBM Systems Journal, Vol 43(1), 2004.

[8] M. Burgess: *An Approach to Understanding Policy Based on Autonomy and Voluntary Cooperation*; 16th IFIP/IEEE International Workshop on Distributed Systems Operations and Management (DSOM), LNCS 3775, Springer, pp. 97-108, 2005.

[9] M. Burgess: *An Introduction to Promise Theory, Part I*; Presentation Slides, NMRG 2006, pp. 1-15, 2006.

[10] M. Burgess: *The Promise of Self-adapting Equilibrium (Keynote speech)*; 5th IEEE International Conference on Autonomic Computing (ICAC 2008), Chicago, Illinois, U.S.A, June 2008.

[11] M. Burgess, S. Fagernes: *Laws of Human-Computer Behaviour and Collective Organization*; IEEE Transactions on Network and Service Management, 2008.

[12] M. Burgess, T. Schaaf: *Integrating cfengine, ITIL, and Enterprise Processes*; HIO/LMU document, September 2008.

[13] EU IST TEQUILA: *Traffic Engineering for the Internet at Large Scale*; (Accessed:) December 2008, www.ist-tequila.org.

[14] The Globus Alliance: Towards Open Grid Services Architecture; December 2008, http://www.globus.org/ogsa/.

[15] M. Göhner, M. Waldburger, F. Gubler, G. Dreo Rodosek, B. Stiller: *An Accounting Model for Dynamic Virtual Organizations*; 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007); Rio de Janeiro, Brazil, May 2007, pp. 1-8.

[16] Hasan, P. Racz, B. Stiller: *Monitoring of SLA Compliances for Hosted Streaming Services*; to be published, 11th IFIP/IEEE International Symposium on Integrated Network Management, New York, June 2009.

[17] Leibniz-Rechenzentrum (LRZ): *LRZ Grid Portal*; http://www.Grid.lrz.de/en/overview.html, December 2008.

[18] J. R. Loyola et al: *A Methodological Approach towards the Refinement Problem in Policy-based Management Systems*; IEEE Communications Magazine, October 2006.

[19] M. Matthias, S. Wesner: *GRID Activies at HLRS*; Presentation Slides, 6th HLRS Metacomputing and GRID Workshop, Stuttgart, Germany, pp. 1-32, April 2003.

[20] Office of Government Commerce (OGC): *ITIL*; http://www.ogc.gov.uk/guidance_itil.asp, June 2007.

[21] Office of Government Commerce (OGC): *ITIL Continual Service Improvement*; The Stationery Office Books, May 2007.

[22] Office of Government Commerce (OGC): *ITIL Service Design*; The Stationery Office Books, May 2007.

[23] Office of Government Commerce (OGC): *ITIL Service Operation*; The Stationery Office Books, May 2007.

[24] Office of Government Commerce (OGC): *ITIL Service Strategy*; The Stationery Office Books, May 2007.

[25] Office of Government Commerce (OGC): *ITIL Service Transition*; The Stationery Office Books, May 2007.

[26] Outsourcing Whitepapers: *The Five Critical SLA Questions: What You Need to Know Before You Define Your Managed File Transfer Service Level Agreements*; Sterling Commerce Whitepaper, August 2007.

[27] A. Paschke, E. Schnappinger-Gerull: *A Categorization Scheme for SLA Metrics*; Multi-Conference Information Systems (MKWI06), Passau, Germany, GI Lecture Notes in Informatics, Vol. 80, 2006.

[28] D. Quan, O. Kao: *On Architecture for SLS aware Workflows in Grid Environments*; 19th International Conference on Advanced Information Networking and Applications (AINA'05), pp. 287-292, 2005.

[29] J. Quittek, T. Zseby, B. Claise, S. Zander: *Requirements for IP Flow Information Export (IPFIX)*; RFC 3917, October 2004.

[30] B. Stiller, D. Hausheer (Eds.): *Development of a Multi-provider Model, an SLM Model, and Service Provisioning Concepts*; EMANICS Project Deliverable D8.2, pp. 1-61, July 2007.

[31] B. Stiller, D. Hausheer, G. Schaffrath (Eds.): *Definition of Service Provisioning Goals, Economic Impacts, and SLA Management Tasks*; EMANICS Project Deliverable D8.1, pp. 1-114, June 2006.

[32] M. Waldburger, M. Göhner, H. Reiser, G. Dreo Rodosek, B. Stiller: *Evaluation of an Accounting Model for Dynamic Virtual Organizations*; Journal of Grid Computing, Springer, Vol. Online First, DOI: 10.1007/s10723-008-9109-9, pp. 1-19, Netherlands, September 2008.

[33] M. Waldburger, B. Stiller (Eds.): *Definition of a Draft Extended IP Network Management Model*; EMANICS Project Deliverable D8.3, pp. 1-68, January 2008.

12 Abbreviations

A4C	Authentication, Authorization, Accounting, Auditing, and Charging
AA	Authorization Authority
AAA	Authentication, Authorization, and Accounting
ABC	Activity-based Costing
ALTO	Application-layer Traffic Optimization
AR	Access Router
AS	Autonomous System
ASAM	Auditing of SLOs Across Multiple Provider Domains
ASP	Application Service Provisioning
AURIC	Auditing Framework for Internet Services
BDIM	Business-driven IT Management
BP3EM	Best Practices, Processes and Promises in Economic Management
BR	Border Router
CI	Configuration Item
CIO	Chief Information Officer
CMDB	Configuration Management Database
COBIT	Control Objectives for Information and Related Technology
CRM	Customer Relationship Management
DiffServ	Differentiated Services
DVO	Dynamic Virtual Organization
DVD	Digital Versatile Disk
DX.Y	Deliverable X.Y
ES	End System
eTOM	Enhanced Telecom Operations Map
FCAPS	Fault, Configuration, Accounting, Performance, and Security
GridAcc	Grid Accounting
ID	Identification/Identity/Identifier
IELOS	Impact Analysis of Economic and Legal Objectives on Basic and Value-added Services
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	IP Television
ISP	Internet Service Provider
IT	Information Technology
ITIL	IT Infrastructure Library
ITILv2	IT Infrastructure Library Version 2
ITILv3	IT Infrastructure Library Version 3
ITSM	IT Service Management
LMU	Ludwig Maximilians University
LRZ	Leibniz Rechenzentrum, Leibniz Supercomputing Center
LSP	Label-switched Path
MeSA	Measured Signaling for Auditing
MPC	MeSA Probe Collector
MPG	MeSA Probe Generator
MPLS	Multi-protocol Label Switching
NAS	Network Attached Storage
NM	Network Manager
NP	Network Provider

OGC	Office of Government Commerce
OGSA	Open Grid Services Architecture
OLA	Operation Level Agreement
P2P	Peer-to-peer
PDCA	Plan-Do-Check-Act
PHB	Per-hop Behavior
PIM-SM	Protocol Independent Multicast-Sparse Model
PoP	Point of Presence
PRIPOL	Pricing by Policies
QoD	Quality-of-Devices
QoS	Quality-of-Service
RSVP	Resource Reservation Protocol
RTT	Round Trip Time
SAN	Storage Area Network
SaPDoGS	SLA and Promise Descriptions of GRID Services
SCTP	Stream Control Transmission Protocol
SID	Session Identifier
SIP	Session Initialization Protocol
SISL	Service Information Specification Language
SLA	Service Level Agreement
SLAPN	SLA Planning and Negotiation
SLI	Service Level Indicator
SLM	Service Level Management
SLO	Service Level Objective
SLO CM	SLO Compliance Monitor
SMONA	Service Monitoring Architecture
SmoothIT	Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies
SOA	Service-oriented Architecture
SP	Service Provider
TCAS	Traditional Cost Accounting System
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTS	Trouble Ticket System
UC	Underpinning Contract
UDP	User Datagram Protocol
VO	Virtual Organization
VoD	Video on Demand
VoIP	Voice over IP
WP	Work Package
WG	Working Group

13 Acknowledgements

This deliverable was made possible due to the large help of the WP8 team of the EMANICS team within the NoE, which includes this deliverable's authors as indicated in the document control. Many thanks to all of them.

14 Annex 1: Promise Theory Workshop Report

- October 21-28, 2008
- Sponsored by EMANICS
    - Mark Burgess, HIO
    - Remi Badonnel, INRIA
    - Jan Bergstra, University of Amsterdam, NL
    - Alva Couch, University of Tufts, USA

This EMANICS workshop on Promise Theory was organized by the Oslo University College. It took place at the Faculty of Engineering (Oslo University College) and at the Oslo Innovation Centre (Research Park), where many technology companies are located. This includes the recently formed Cfengine AS spin-off company, based on the promise theory developed at HIO.

The workshop aim was to gather theoretical and practical experiences related to Promise Theory and allow the key researchers in the area to discuss topical issues together. Promise theory defines a framework for describing and analyzing policy governed services in the context of fully autonomous agents, managing expectation, and understanding the economics of these interactions. Agents assist one another in purely a voluntary manner, and they use promises to manage their expectations of one anothers' behavior.

The organization of the workshop was similar to a Dagstuhl seminar including several presentations, tutorials and extensive discussions, reflections and interactions amongst participants. The workshop began with a tutorial about cfengine 3, the reference implementation of promises as a technology for server management. Later discussions pointed out the multiple challenges and opportunities offered by the concept of promises. From a theoretical viewpoint, the meeting covered multiple topics including the issues of autonomy in network and system management, the concepts of intentions, promises and deceptions, the analysis and extraction of emergent behaviors based on promise graphs, and the modeling of promises using topic maps. From a technological/practical viewpoint, we have given paid particular interest to the instantiation of Promise Theory by the autonomic maintenance system Cfengine. The new version of Cfengine introduces a new extended language that provides support for specifying and keeping promises with self-healing technology.

Further topics for discussion included the predictions made by promise theory concerning logical and economic reasons for architecture (centralized versus distributed organizations, selection and redundancy of policy servers), control issues (to what extent the behavior of Cfengine agents can be determined based on their context) and scalability issues (network patterns), and the use of semantic type content of promises is forming knowledge networks.

15 Annex 2: Business Driven IT Management (BDIM) Workshop Report

- Workshop at the USENIX/LISA conference
- San Diego, November 9, 2008
    - Organized by Mark Burgess, HIO
    - Claudio Bartolini, HP Labs.
    - Sponsored by EMANICS

About 20 people attended from a variety of enterprise and government organizations across the US came to discuss the ways in which IT management can be aligned with the business and economic goals of organizations. Jacques Sauve (Brazil) who organized the first IEEE workshop together with Claudio Bartolini was also present.

The format of this workshop was a round-table discussion of a number of key questions posed by the organizers concerning the meaning and importance of aligning IT services with business goals. There was consensus that the workshop was most interesting. A brief summary follows.

BDIM — What's it about?

The following topics were mentioned:

- ITIL - “Best practices”
- Security
- Productivity
- Promises (SLA etc.)
- Configuration management
- Handling diversity
- Goals — aligning with business goals
- Business processes?

The following questions were posed to kickstart the discussion.

- Do best practices exist?
- What metrics do we have for BDIM?
- What are business processes?
  - → Sales/Admin
  - → Research/Develop
  - → Production
  - → E-commerce
  - → Communications (phone/mail etc)
- Mission critical - what does it mean?

Ideas from around the Table

Best practices are always context dependent. Many organizations around the table were thinking in terms of ITIL and service management. In particular, the idea of a service catalogue was a useful way to document activities and responsibilities within organizations. There was however consensus that the “Service Level Agreement” (“one size fits all”) concept was a poor tool for managing services. In most cases SLAs are not true agreements (i.e. two or more parties have signed a contract), rather they are service level promises provided on a take-it-or-leave-it basis.

Jacques Sauve suggested that the SLA is too small a keyhole to see business through. There are broader aspects of business than atomic services. For all of its flaws, however, the meeting did little else than talk about SLAs so it was clearly an important concept for semi-formalizing service management. Often even the SLA itself was considered “bogus” it was valued nevertheless for its trickle-down effect on communicating intent to parts of the organization. The SLA was seen as a tool for focussing minds rather than actually managing service levels. It was felt that the idea of a promise (in the sense of promise theory) was in fact a good abstraction of the positive aspects of the SLA. One useful feature

of the SLA, which also applies in the case of “compliance” like is that it makes someone accountable for keeping promises. Without such a tool there is no clear source to point to in case of inadequacy. Another view was that an SLA should be considered as defining the limits of “engineering tolerance” for the services being provided, i.e. something that tells us when to stop improving, or when the service is good-enough -- so that we do not have infinite cost associated with a promise.

Agility and Homogeneity

It was questioned whether it is indeed *practical* to align IT with business goals in a meaningful sense. The business mission changes too often, and IT departments have to keep up! This kind of agility was considered important to everyone regardless of whether they belonged to government or industry. In some cases (*e.g.*, Google) there is no importance to run-time state; all assets can be regenerated quickly so there is no need for stability. In other cases (*e.g.*, banking) there is no room for instability. Often companies cite Service Levels that are pointless, *e.g.,* five-nines reliability (99.999% uptime). Is this useful, or do we actually prefer 99% with agility for change? It was pointed out that optimization of a process generally makes it rigid and inflexible, so that optimization is a double edged sword. How should one view failure to comply with an SLA target (a service promise)? It was suggested that a well designed SLA allows a provider to fail to deliver and pay a penalty without significant loss. The converse is that an SLA allows us to “just underperform”. The average time to roll-out could be used as a metric for agility of an organization. The latency for a change to come into play is an alternative metric. Agnosticism of operating platform allows agility.

Enclaves of Uniformity?

A compromise between complete homogeneity and general heterogeneity is to allow a few types (*e.g.*, small, medium, large). Virtualization is a way of managing non-uniformity today. Cfengine and voluntary cooperation are about the most realistic model for change and configuration management. Complexity has a cost in support, but some tools make this cheaper than others. Many organizations distrust variations. Can we trust heterogeneous systems? A big problem is comprehension: how can we explain to management what the system will do if it is too complicated? We need to be able to understand business needs much better to be able to justify heterogeneity to management. System homogeneity is the “nuclear weapon” of predictability in server management. It is not necessary to make all machines alike, but it is a common model because traditional technologies are poor in maintaining variations.

Cost of Management?

There are many costs in IT management. Upgrading is a potentially destructive procedure, while not-upgrading can be a potential threat to security. Compliance with external requirements can be highly costly (SOX etc). Government institutions in the US have about 300 windows settings that are required by law. These often interfere with the need to carry out work on the computers. One ends up trading the risk of “security breach” against the risk of “getting work done”.

The cost of having “incident” tickets is high — how can we understand this and proactively prevent the incidents from occurring. Again the self-healing maintenance approach seems to be the answer. The cost often manifests itself further down the road so is apparently hidden to management until too late.

A significant challenge for businesses lies in scaling up, or transforming from a small startup into an enterprise. Bureaucracy is an easy trap which can smother an enterprise

and stifle innovation (the opposite of agility). Too many procedures binds effort to overhead instead of agile production.

Business is considered to be very different from engineering, but should it be? Can one consider the running of a business as a type of engineering? Yes, but there is a cultural gap between engineers and MBAs so they talk quite different languages. Make the connection between business thinking and tech thinking is a key challenge. There needs to be mutual understanding. Tech must learn from business and vice versa.

Compliance, *e.g.*, SOX/EUROSOX

What is the actual function of these laws? US participants felt that the law was useful indirectly as it forces companies to represent themselves honestly. The law makes the CEO legally responsible, but the effect is to improve the “technical standard of reporting” throughout the organization. The trickle-down effect is important Name one person, but he holds others responsible. Is this the same basic argument for effect as with the SLA? The fact that is points a finger of responsibility. Promise theory suggests that this is indeed the same. Compliance should simply be viewed as a service promise on the part of the company even though the law is an “obligation”.

Control versus Voluntary Cooperation

The issue of obligation was brought up as the standard by which most organizations are managed. Leaders make the decisions on others' behalf usually, and yet this is usually a bad model for making a system work since the understanding of technical information is often slow or poor by management. The voluntary cooperation model seems to be a more useful model because it managed expectations better. Better to see what the components can promise than make unrealistic demands.

Another area in which promise theory was appreciated by the attendants was in scaling huge server parks, with 35,000 machines. The main obstacles turned out to not be technical but political (getting everyone to work cooperatively towards a common goal). Many compliance rules that come from above are not based on realistic expectations. Tech people need to inform and communicate back what is reasonable more effectively to management. Not only business driven IT, but IT informed business.

Knowledge and Data Mining

Knowledge management was proposed as an important aspect of IT management in the future. This is not about ontologies for context adaptation of pervasive computing policies, but also about the integration of enterprise knowledge for humans to better understand the intention behind policy. Can IT show which customer practices lead to profits or predict which might do in the future? Communication between business and IT is very important for synergy. We need to look at the payload of transactions flowing through a system to understand what is going on, not just count numbers or look at performance. Semantics and content allow us to track and understand the meaning.

Policies that are good for business are probably dependent on the economic reality of an organization. What phase is the company currently in? In an expansion phase, we are trying to push out services as fast as possible and performance optimization is a goal. In a recession or contraction phase, optimization is about cost cutting.

Summary

The service paradigm is simple and useful, but sometimes also simplistically applied. There is presently too much focus of simple slogans like “five nine's reliability” without rational motivation.The SLA is the only tool people recognize -- but SLA is the wrong term, “service

promise" is better, because so-called SLAs are not always agreed upon by all parties. Is there something else we should be looking at, *e.g.*, cooperative promises?

For business alignment, user or customer experience is the key driver. We should look for metrics for this. Service level management seems more important than the SLA itself (i.e. the SLA is like a flag to rally the armies of the organization), but the processes it unleashes are mainly in someone's head, not documented so often the only thing written down is the SLA. This needs to change to make an organization effective. We have to educate everyone in IT about business processes, and where value is created, and inform business leaders about IT capabilities so that decisions are being made properly.

16 Annex 3: 72nd Internet Engineering Task Force Meeting (IETF 72)

- Date: July 27 - August 1, 2008
- Dublin, Ireland
- Participants: 1300+
- Participant from UniZH: Fabio Hecht

ALTO BoF

The 72nd IETF Meeting took place in a hotel in the far west of Dublin, with more than 1300 attendees. Seven meetings took place in parallel, in a total of three sessions per day, in six days. Attendance in the ALTO BoF meeting is estimated at about 300 participants.

Problem Statement

The meeting started with a presentation about the problem (Enrico Marocco, Vijay K. Gurbani). The focus was on P2P traffic. Some approaches were presented, and a line was drawn between peer-to-peer (resource-based) and Client/Server (host-based). The statement was presented. It was argued that peers performing measures are expensive and not accurate. Mentions throughput as a special case – wondering whether he will be able provide a solution for that. The purpose of this BoF is to try and become a WG in the IETF to define only the protocol – not caring about metrics, for example. He mentioned privacy and security aspects but simplified the solution by stating that the service is optional. The current draft can be obtained from http://tools.ietf.org/html/draft-marocco-alto-problem-statement, which contains what has been discussed on the mailing list.

After the presentation, many questions popped up form the audience. Reducing inter-domain traffic is THE problem to solve? Not the only, but if this is about problem statement, it must be very clear what the main problem is. Another point is about this specific routing information (intra-domain vs. inter-domain): is it already available to overlay applications? Some say it can be figured, but still there should be a standard way to fetch it, not a hack. Another person points out that optimization is a balance, and maximizing intra-domain traffic may not be ideal – what are they trying to optimize? A more experienced engineer from Verizon mentioned that access network is usually the bottleneck. Also, application scenarios are a must. Although the protocol should not focus on the metrics, they are important to be foreseen in order to justify the effort to define it.

ALTO Survey

The next presentation by Volker Hilt dealt with a survey, available on http://tools.ietf.org/html/draft-hilt-alto-survey. He started by showing examples on how to select a good peer.

He presented several solutions, such as IDMaps, Vivaldi, iPlane, Ono Project, P4P. After the presentation, there were many questions, some of the challenging the whole mechanism. One of them stated that nowadays the ISP don't necessarily own its facilities, they rent it. The ISP does not own its domain. Also, a provider may cover several countries – intra-domain won't necessarily mean low cost and high quality. Another interesting question was: would all of those presented approaches fit into whatever will be defined? Since there are many metrics, how to keep semantic information consistent?

Distance-related Network Costs

The presentation of Henning Schulzrinne was mainly about costs and pricing. He showed the current cost of bandwidth to be about 0.15 US$/Mbps/month. Downloading a DVD would cost approximately 1.05 US$ to an ISP. He presented a discrete step graph, picturing cost proportional to distance. Then he mentioned alternative charging schemes, for example, a client losing priority after hitting a certain limit, or having free local traffic (potentially limited), or varying according to time of day (at nights faster), and others. The bottom line was that the economic decisions should be taken by the customers. As usual, there were many highly critical questions. One participant stated that the highest cost is backbone cost, doesn't matter if traffic goes long or short, if uses the backbone, it costs. Then, a discussion about what a backbone is took place. Another interesting question challenged whether the price should be proportional to the cost.

Requirements

Sebastian Kiesel's presentation focus on how the system could work in a tracker- and DHT-based P2P system. An ALTO server may have to implement several interfaces: feeding topology info, providing coordination between ALTO servers, besides the client interface. He presented the requirements for the protocol, that can be viewed in detailed in http://tools.ietf.org/html/draft-kiesel-alto-reqs. Finally, he showed some examples of criteria: distance and cost, but it is still unclear what metrics shall be used. Questions regarded caching and lifetime of results, peer selection process that can be bidirectional (providing peer also performs selection), and issues related to ISPs caching content.

Caching and Peer Selection

The presentation of Reinaldo Penno was about cache discovery – not strategies. He presented some techniques, for example using DNS or the ALTO server. Here, the meeting got controversial. Should the ALTO server help the peer-to-peer application to find content or caches? If that is the case, there are already many resource discovery protocols, why not reuse an existing protocol instead of creating a new one for this proposal?

Similar Problems in Multi-Homed Networks

The last presentation by Dimitri Papadimitriou targeted the assumption that, in the future, more and more content will be available through multiple sources – be them in peer-to-peer networks, CDNs. Multiple sources mean multiple paths, and even multiple protocols can be available (*e.g.,* IPv4 and IPv6). Dimitri showed that those applications would need to select the best paths to reach a particular content, and the ALTO server can play a role. He showed current solutions and proposed one of his own. A detailed overview of his presentation can be obtained from http://www.ietf.org/internet-drafts/draft-bonaventure-informed-path-selection-00.txt.

Charter discussion

After all the presentations, there was a general discussion about diverse subjects. One of the most discussed points regarded service discovery – some people were meaning

discovering the ALTO server and some people were thinking about using the ALTO server to obtain alternative sources, such as caches or additional peers. It was more or less agreed that they shouldn't reinvent the wheel, and use one of the many standards already in place for such a role. Also, why put together provision and discovery of service? Generally speaking, separation is a good thing.

There was also criticism about the vagueness of the problem statement. It should be short and to the point. Also, the requirements proposed already state the solution. That is very biased.

At the end, the chairs requested the audience to hum if they were in favor of continuing the work creating an ALTO WG. Many people agreed, shyly. Then, people were requested to hum if they were against it. This time, less people hummed, but they did it vehemently. There were several arguments against the standardization of the protocol. Everyone agrees that this is a problem to be solved. However, there are people who are not convinced this is the best solution, since it is not clear that peers get a better answer by trusting the ISPs than they can figure out by themselves; or doubt the approach is the best possible. On the other hand, there are many separate technologies popping up to set up ISP-assisted peer selection, and a standard would be desirable, otherwise there will be many incompatible solutions. The fact is that there is a lot of research work involved and that is not the purpose of the IETF. It might be too early to define a standard, and research – like SmoothIT – must be done to prove the point to all the community.

17 Annex 4: 73rd Internet Engineering Task Force Meeting (IETF 73)

- Date: November 16-21, 2008
- Location: Hilton Hotel, Minneapolis, MN, USA
- Participants: 1115 registered
- Participant from UniZH: Fabio Hecht

ALTO – Application-Layer Traffic Optimization WG

The newly created ALTO WG is attracting great interest from the IETF. An estimated number of 240 persons attended the meeting.

The first talk was a short introduction, by the chairs, that presented the last important modifications in the charter, stressing important changes. The most important ones are the following:

- focus is peer selection only;
- goal is to perform better than random peer selection – not optimal.

After this talk, Aaron Falk talked about the IRTF p2prg (Peer-to-Peer Research Group). They are creating a new charter and are looking for chairs.

The next talk, by Enrico Marocco (chair, Telecom Italia), presented the latest updates in the problem statement draft, reviewing discussions since Dublin. The document already points the solution: “a topology information (...) will allow applications to improve their performance and will help ISPs make a better use of their network resources”. It is very clear that the objective of the WG is to improve performance and reduce inter-domain traffic, focusing on localization of traffic. They say it can also be

something other than localization, but it is not clear what can that be. The original document contained the word “oracle”, referring to the solution of Feldmann et al., but due to requests the term has been changed. The first comment from the audience was that solutions should not be part of the problem statement – an analysis of the solution space must be done in the first place. Another concern regards privacy. The document states that “the application does not have to disclose information it may consider sensible”, but it is actually very difficult to say the least to determine what is private and what is acceptable.

Sebastian Kiesel presented his draft on requirements. He showed changes in the document since the Dublin meeting, the most important ones being the following:

- not seeking the optimal solution, just better than random one;
- removed the suggestion of a sorting oracle, that would be not appropriate for requirements;
- define core set of attributes for expressing preference, extensible to other ones.

The document reads much like the one on problem statement, and a long discussion about this was started, including the theme of why a requirements document is useful after all. In the end, it was more or less agreed that the requirements should be revisited after the problem statement document is refined.

The next talk, by Richard Yang (P4P), was entitled “P4P Design and Implementation”. They are working with two services: location and pDistance. Location is stable and returns a PID for each peer. One PID groups peers that are close together, and the granularity can be played with (AS level is suggested). The pDistance service returns a distance between two peers. PIDs can be inter-domain or inter-domain, and the return value can be ordinal, or numeric (he prefers the latter). Types of metrics can

be: hopcount, air-mile, cost (which is the default). A person in the audience asked why then not use one of the existing Internet maps (“looking glasses”) that are available, and how different would it be to just use them. This is a point that is still controversial.

Richard Woundy presented then the talk Comcast's Experiences In a P4P Technical Trial, in which he showed some details about the draft with same title. They are working directly with P4P and ran a test with their customers. The results were improved download speed, and localization (less inter-domain traffic). The experiment involved a single, 21MB, file. It was a Pando client update, so users were forced to download it. Criticism to this experiment are:

- the file is small in comparison to what people have been trading in file sharing applications;
- it was a forced download, so the gains are maximized due to a large swarm, which is not always the case.

The next presentation was about the draft “ALTO Information Export Service”, which suggests that the clients download a table containing full preference information from the ISP so they do not have to keep on constantly querying the ALTO service. He argues that the P2P applications can already do a pretty good job figuring out routing information. What is missing is only the ISP preference, which can be described in a small enough table to be downloaded completely. The presentation included the format of the file, as contained in the draft, with three fields per record: designator (“asn” or “cidr”), AS number or IP prefix (CIDR), and a priority. In his example, the application would sort the peers in three sets: preferred, default, and to be avoided. The size of the table in the tests were 1.5Mbytes (compressed). Possible issues recognized by the author are the redistribution of

information by peers (could produce outdated information) and, service discovery. This work is being carried on by BitTorrent.

Stefano Previdi (Cisco) presented the next talk, entitled Routing Proximity. His goal is to establish metrics to be classify peers with regard to proximity. Routing databases (ISIS/ OSPF/BGP) have already proximity metrics, so they can be used for the purpose of calculating the distance between peers. He believes everything needed to achieve localization is already available. ALTO should be just an interface to support routing (*e.g.,* BGP) information to overlay. An important question is the addition of other metrics, such as cost, link capacity, and congestion – term that appears as a prominent motivation in the charter text.

The last presentation, “A Multi Dimensional Peer Selection Problem”, by Saumitra Das, discussed the fact that many different factors influence peer selection. Some information might come from ISPs and but ultimately the peers make the selection using information such as reputation. He suggests that different types of ALTO servers (*e.g.,* P4P) can coexist.

The meeting ended with a word from the chairs to keep up current work and discussion on the mailing list.

LEDBAT – LEss Than Best Effort Transport WG

LEDBAT is the newly created WG that stems from the TANA BoF. The name was changed since some people considered the term “Advanced Network Applications” too unspecific. Just like ALTO, it aims at coping with P2P traffic, but works at the transport layer. The idea is to design a protocol that does not interfere with regular (best-effort) traffic, utilizing unused bandwidth. Moreover, the protocol would be able to “scavenge” network resources that would otherwise be unused.

Stanislaw Shalunov (BitTorrent) opened the session with a charter recapitulation. The objective is “to standardize a congestion control mechanism that should saturate the bottleneck, maintain low delay, and yield to standard TCP”. It originated at the P2PI workshop at MIT in May/2008, creating the ALTO and TANA BoFs in Dublin, which lead to the ALTO and LEDBAT WGs in Minneapolis. The problem being solved is that TCP fills router buffers if congested, and the buffer can be large, the likely worst case are home uplinks. Since most traffic on home uplink is P2P-related, it leads to a delay in other applications that users might be using. There are two work items: experimental congestion control and current practices of applications (using multiple connections). Applications (BitTorrent, web browsers, mail servers) create multiple connections to try and maximize throughput and add stability. The practice is common, but considered more a poorly documented hack. The idea is to research and document how applications use multiple connections. Bob Briscoe mentioned that they should take into consideration how to respond to the congestion. Another person asked whether the protocol should “yield to TCP” or to be something better, more modern than TCP.

Satish Raghunath (Juniper.net) presented the next talk, entitled “LEDBAT App Practices and Recommendations”. He mostly talked about why current P2P applications open multiple connections. Mentions reliability that comes with diversity, but comes with overhead, and impact delay-sensitive traffic, due to more state needed within TCP termination devices and middleboxes. The applications try and find a “right” number of connections – too few or too many can cause problems – the objective is to maximize download speed, for example, in BitTorrent. The objective is to elaborate a document with recommendations to the number of connections. The audience asked whether he will/did

look also at UDP or only TCP? At the moment, only TCP, someone (!) could contribute with a draft. Another person pointed out that Firefox opens up to 16 connections to each host, is that a good or a bad number? People also pointed out that is not true that opening multiple connections always makes downloads faster. They experimented with the iPhone over 3G and stated that handshake and congestion control don't work, especially for small images it may not be worth to open another connection. Next step: do the research!

Murari Sridharan (Microsoft) presented next “Low priority TCP: Receive-Window Control”. It was a paper presentation about BATS (background transfer service). They adapt the receiver window to create a low priority service, and he explains how. The algorithm has 2 modes: rate limiting mode (1) and window scaling mode (2). Mode 1: gets accurate RTT samples, mode 2: uses binary search to drive towards target window, assuming the value lies between Wmin and Wmax; depending on whether there

is congestion, window size is adjusted. Bottom line: it maintains low delay and yields to TCP. It requires no support from the network, although additional information helps it adapt quicker. He suggests this work as a starting point of how LED could look like. Question 1: has he analyzed if it works with competition between several connections? The presented answered he is working on getting these numbers. Other important questions regarded RTT independence and whether the background flow should be starved if necessary, and respective answers are that it “tries” to be independent from RTT (whatever this means) and that starvation can be controlled.

The next presentation was very short – only 5 minutes – by Nick Weaver (Berkeley), entitled “A Couple Academic Thoughts on LEDBAT”. He affirmed that there are two separate problems without use of packet marking or AQM (active queue management): detect buffer occupancy problems and detect and yield in common congestion to other types of traffic. In his opinion, LEDBAT should be defined as a TCP operating mode. Only one side should need to use the defined congestion control policy (à la 4CP) in order to ease deployment issues. He raises the question on whether DiffServ marking should be used. Bob Briscoe thinks that marking would not work, because it doesn't matter what they put there the operator won't believe. The point is that they would be marking them to be low priority, not to get higher priority.

Stanislaw Shalunov (BitTorrent, chair) presented “Low Extra-Delay Background Transport”, his idea of what could be standardized by LEDBAT. In his view, the main problem is that TCP fills the buffer and it can be large, introducing high delay in case of congestion. This delay breaks real-time applications like VoIP when a P2P application (like BitTorrent) is running. It also slows down considerably traffic that is not real time, like web browsing. He raises the question of how large should the buffer be, but does not have the answer. This raises a long discussion, and Stanislaw mentions measuring one-way delay, which is deemed impossible by at least some of the audience. He presents details of his approach, which includes using smaller packets and estimating queue delay in order to reduce window size before packet loss occurs. The presenter states though that he has done it and tested in BitTorrent DNA by 7M active users. The audience asks whether he has any numbers to show and confirm his statements, but he has not. Part of the audience did not really agree that using small packets should help, the reality is that packets are of a small size for a long time. The author says that he uses smaller packets in order to minimize serialization time and be able to obtain faster speed in a slow link.

The meeting ended with a word from the chairs pointing the WG to future work. They envision researching how applications use multiple connections in order to maximize their download speed and plan to further refine current studied approaches.

18 Annex 5: Selected Cooperation Work

This section of selected cooperation work covers a range of work started recently in the context of EMANICS WP8 as well as sometimes shortly before. Thus, the content in this section consists out of paper abstracts and summaries from single institutions (early starts) as well as joint work between EMANICS partners (recent starts).

To provide an overview of these areas of work, the following subsections list authors, abstracts, and the table of content, if they exist, or a respective sketch of the idea to be worked on in the next month of EMANICS. In case of a full paper being available, it is part of this deliverable at its final end, following the sequence as their summaries below.

Thus, D8.4 covers 2 single affiliation papers published (including one key note) and 4 joint affiliation papers being published (including one set of workshop proceedings), totalling in 6 papers published (including one accepted journal paper).

18.1 Monitoring of SLA Compliances for Hosted Streaming Services

Authors:

Hasan, Peter Racz, Burkhard Stiller, University of Zurich

Abstract:

Monitoring of Service Level Objectives (SLOs) determines an essential part of Service Level Agreement (SLA) management, since customers are to be reimbursed, if a provider fails to fulfil them. By automating this process, a timely detection of a violation is possible. The compliance approach must be flexible to adapt to potential changes, must be scalable with respect to the amount of data, and has to support multi-domain environments. This paper determines a scenario and defines relevant SLOs. Key requirements are derived, the respective architecture is designed, and the approach is implemented prototypically based on a generic auditing framework. Furthermore, a new scheme is proposed that considers the degree and duration of SLO violations in calculating reimbursements.

Table of Contents:

Publication:

Hasan, Peter Racz, Burkhard Stiller: *Monitoring of SLA Compliances for Hosted Streaming Services*; to be published in: 11th IFIP/IEEE International Symposium on Integrated Network Management, New York, June 2009.

18.2 Evaluation of an Accounting Model for Dynamic Virtual Organizations

Authors:

Martin Waldburger, Burkhard Stiller, University of Zürich
Matthias Göhner, Gabi Dreo Rodosek, Universität der Bundeswehr München
Helmut Reiser; Leibniz Supercomputing Centre

Abstract:

Accounting of Grid resource and service usage determines the central support activity for Grid systems to be adopted as a means for service-oriented computing in Dynamic Virtual Organizations (DVO). An all-embracing study of existing Grid accounting systems has revealed that these approaches focus primarily on technical precision, while they lack a foundation of appropriate economic accounting principles and the support for multi-provider scenarios or virtualization concepts. Consequently, a new, flexible, resource-based accounting model for DVOs was developed, combining technical and economic accounting by means of Activity-based Costing.

Driven by a functional evaluation, this paper pursues a full-fledged evaluation of the new, generically applicable Grid accounting model. This is done for the specific environment of the Leibniz Supercomputing Centre (LRZ) in Garching, Germany. Thus, a detailed evaluation methodology and evaluation environment is outlined, leading to actual model-based cost calculations for a defined set of considered Grid services. The results gained are analyzed and respective conclusions on model applicability, optimizations, and further extensions are drawn.

Table of Contents:

Publication:

Martin Waldburger, Matthias Göhner, Helmut Reiser, Gabi Dreo Rodosek, Burkhard Stiller: *Evaluation of an Accounting Model for Dynamic Virtual Organizations*; Journal of Grid Computing, Springer, Vol. Online First, DOI: 10.1007/s10723-008-9109-9, pages 1-19, Netherlands, September 2008.

18.3 Customer Service Management for Grid Monitoring and Accounting Data

Authors: Timo Baur, LRZ/MNM (LMU), Samah Bel Haj Saad, UniBwM

Abstract:

Experiences with the management of Grid specific monitoring and accounting data have shown that current approaches do not sufficiently support a distinction between providers, users and customers of a Grid. This gap can be filled by the use of Customer Service Management techniques which enable customers to individually monitor and control their subscribed services. We adapt a Customer Service Management scenario to Grid environments and outline an architecture dedicated to the management and visualization of monitoring and accounting data. To proof the concept, a prototype based on standard Grid components which manages user's needs and interactions with the resource provider is presented.

Table of Contents:

  - → CSM Basic Service
  - → CSM Application
- Conclusions
- References

Publication:

Timo Baur, Samah Bel Haj Saad: *Customer Service Management for Grid Monitoring and Accounting Data,* Lecture Notes in Computer Science, Vol. 4785, Springer 2007, pages 216-228, 18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2007), San Jose, California, U.S.A., October 2007.

18.4 Integrating cfengine, ITIL, and Enterprise Processes

Authors: Mark Burgess, HIO, Thomas Schaaf, LMU

Abstract:

cf. full text of document

Table of Contents:

- Introduction
- Cfengine past and present
- ITIL past and present
- A meeting of mind-sets
- Using cfengine to implement ITIL objectives
- Summary
- ITIL terminology

Publication:

Mark Burgess, Thomas Schaaf: *Integrating cfengine, ITIL, and Enterprise Processes,* HIO/ LMU document, September 2, 2008.

18.5 Joint EC-GIN, EMANICS, and SmoothIT Workshop on "Economic Traffic Management" (Proceedings)

Authors: Burkhard Stiller (Edt.), UniZH

Abstract:

The Workshop on "Economic Traffic Management (ETM)" had been jointly organized by the European research projects EC-GIN, EMANICS, and SmoohtIT. The common denominator of these three projects can be found in the topic of economic management, which includes the question, whether economics and economic theory is applicable in network management in general, in which way this will be beneficial compared to traditional network management approaches, and which players will benefit from such an approach.

Publication:

Burkhard Stiller (Edt.): *Joint EC-GIN, EMANICS, and SmoothIT Workshop on "Economic Traffic Management" (Proceedings)*, IFI Technical Report, No. ifi-2008.10, August 2008.

18.6 The Promise of Self-Adapting Equilibrium

Authors: Mark Burgess, HIO

Abstract:

How should we understand autonomics? As software engineering, as biology? Mark Burgess is the visionary author of cfengine, probably the first autonomic system for server management dating back to 1993, which now manages hundreds of thousands of computers all over the world.

Since writing his manifesto “Computer Immunology” for self-repairing computing in 1998 he has led research efforts at Oslo University College to realize self-healing systems in practice, using strong scientific and engineering principles. In this talk he will share with us some

of the principles that have made cfengine successful and his vision for realizing autonomic computing, including the importance of promise theory and dynamic equilibria that not only offer engineering principles, but also reveal the essential economics behind cooperative

computing.

Publication:

Mark Burgess: **The Promise of Self-Adapting Equilibrium (Key Note Speech)**, 5th IEEE International Conference on Autonomic Computing (ICAC 2008), June 2-6, 2008, Chicago, Illinois, U.S.A.